GRAPH-BASED CONTROL OF NETWORKED SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Meng Ji

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2007

GRAPH-BASED CONTROL OF NETWORKED SYSTEMS

Approved by:

Dr. Magnus Egerstedt, Advisor
Associate Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Chuanyi Ji
Associate Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David Taylor
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Panagiotis Tsiotras
Professor, School of Aerospace Engineering
Georgia Institute of Technology

Dr. Patricio Vela
Assistant Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: March 28th, 2007

观网

纪萌

蚍蜉能撼树，螳臂可挡车，
尘沙观世界，滴水见江河。

To see a world in a grain of sand,

And a heaven in a wild flower, ...

William Blake, Auguries of Innocence

ACKNOWLEDGMENTS

First and foremost, I would like to express my appreciation to my advisor, Dr. Magnus Egerstedt, for his guidance and support, without which this dissertation would not have materialized. Your unquenchable enthusiasm and tireless hardwork have been the most invaluable encouragement to me. I also wish to thank Dr. David Taylor and Dr. Patricio Vela for serving on my reading committee, and Dr. Chuanyi Ji and Dr. Panagiotis Tsiotras for being on my defense committee.

The collaboration with Dr. Mehran Mesbahi and AmirReza Rahmani on the topic of controllability is fruitful and has resulted in an important part of this dissertation. The hierarchical containment control is resulted from the collaboration with Dr. Giancarlo Ferrari-Trecate and Dr. Annalisa Buffa. I also enjoy the meaningful discussions with Dr. Shun-ichi Azuma and appreciate his help in deriving the numerical solution of the assignment problem.

Furthermore, I am pleased to thank my fellow students in the GRITS Lab for their friendship and many meaningful conversations: Tejas Mehta, Florent Delmotte, Henrik Axelsson, Patrick Martin, Shun-ichi Azuma, Abubakr Muhammad, Mohamed Babaali, Jean-Pierre de la Croix, Staffan Bjorkenstam, Mauro Boccadoro, Xu Chu Ding, Anders Gustafsson, Musad Haque, Johan Isaksson, Angela Schoelling, Brian Smith, David Wooden.

Last but not least, I would like to thank my wife and my parents for their endless love and unwavering support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Networked systems have attracted great interests from the control society during the last decade. Several issues rising from the recent research are addressed in this dissertation. *Connectedness* is one of the important conditions that enable distributed coordination in a networked system. Nonetheless, it has been assumed in most implementations, especially in continuous-time applications, until recently. A nonlinear weighting strategy is proposed in this dissertation to solve the connectedness preserving problem. Both rendezvous and formation problem are addressed in the context of homogeneous network. *Controllability* of heterogeneous networks is another issue which has been long omitted. This dissertation contributes a graph theoretical interpretation of controllability. *Distributed sensor networks* make up another important class of networked systems. A novel estimation strategy is proposed in this dissertation. The *observability* problem is raised in the context of our proposed distributed estimation strategy, and a graph theoretical interpretation is derived as well.

The contributions of this dissertation are as follows:

- It solves the connectedness preserving problem for networked systems. Based on that, a formation process is proposed.
- For heterogeneous networks, the leader-follower structure is studied and sufficient and necessary conditions are presented for the system to be controllable.
- A novel estimation strategy is proposed for distributed sensor networks, which could improve the performance. The observability problem is studied for this estimation strategy and a necessary condition is obtained.

This work is among the first ones that provide graph theoretical interpretations of the controllability and observability issues.

CHAPTER 1

INTRODUCTION

A *networked system* is a collection of autonomous agents equipped with a certain degree of sensing, processing, communication, and maneuvering capabilities for its operation. The behavior of such a system depends not only on the dynamics of each individual vehicle, but also on how the network is connected. The agents in the network are abstractions of some autonomous entities, such as mobile robots, intelligent sensor nodes, unmanned air vehicles, autonomous underwater vehicles, spacecrafts, or satellites. The applications include formation flying, collaborative sensor arrays, distributed sensor networks (DSNs), terrestrial planet finder missions, and so on.

1.1 Motivation

The study of networked systems is largely inspired by the collective behavior of biological systems, such as colonies of ants, hives of honey bees, flocks of birds, and schools of fishes. Usually, in these systems, each individual has very limited sensing, communication, and manipulation abilities. However, a well organized large collections of these elementary, simple individuals can produce remarkable capabilities and display highly complex behaviors by following some simple rules which require only local interactions among the individuals.

Inspired by the collective behavior of biological systems, researchers have started focusing their attention on distributed control methods, where each agent makes decisions based solely on local information. The major objective of the research is to achieve better performance by designing a distributed control method which can assemble a number of simple, inexpensive machines rather than a single, complicated, expensive machine.

The control of networked systems has been greatly enabled by the recent developments of powerful control techniques, advancements in computation and communication capabilities, and the advent of other enabling technologies. However, a problem posed to the distributed control of networked system is that in most cases, the sensing and communication capabilities of the agents are bounded by either power constraints or bandwidth limitations. Moreover, the distributed control problem faces the challenge of changing communication network topologies. In other words, the topology of the network depends on the location of the agents and, therefore, changes with the evolution of the network. Such constraints make many traditional coordination methods, which require global information available to each agent, obsolete.

Graphs have been proved to be useful tools for encoding the local interactions and information flows in networks. Aided by algebraic graph theory, this thesis concentrates on how a group of autonomous agents can be coordinated in a distributed way, so that the group can collectively achieve a desired global goal, such as rendezvous, desired formations, or surveillance. Several issues are addressed in this dissertation, including connectedness preserving in homogeneous networks, and controllability and observability of heterogeneous networks.

Throughout this thesis, we assume that N agents constitute the nodes of the network we study, and the state of each agent is given by $x_i \in \mathbb{R}^n$. Their dynamics are given by a single integrator:

$$\dot{x}_i = u_i, \ i = 1, 2, \ldots, N. \tag{1}$$

As such, we are focusing our attention on interaction and high-level control strategies rather than on nonlinear vehicle models, and we assume that the dynamics along each dimension can be decoupled. That is, in some cases, it suffice to analyze the property of a single dimension of all the agents to understand the behavior of the group.

1.2 Background

The history behind networked systems can be traced back to Reynolds' "boids" model [1] and the work by Vicsek et al. [2]. In the last decade, numerous results were obtained [3–13]. This body of research focuses on two main venues, homogeneous networked systems, where agents or nodes are identical to each other, and heterogeneous networks, where some individuals have superior capabilities.

1.2.1 Homogeneous Network

1.2.1.1 Agreement Problem

A considerable amount of research work has contributed to the analysis and implementation of the agreement problem. The agreement problem, in the context of networked system control, is to drive a certain state-dependent variable of each individual agent to a common value, i.e.,

$$x_i(t) \to x_c, \quad t \to \infty, \quad i = 1, \ldots, N. \tag{2}$$

In different circumstances, the agreement problem has be called swarming, schooling, flocking, or rendezvous.

In Reynolds' vanguard "boids" model [1], each agent only reacts to its neighboring flock mates, i.e., those agents within a certain distance. In addition, they have to follow three *ad/hoc* protocols, namely, separation, alignment, and cohesion. As a special case, Vicsek et al. [2] studied the situation where all the agents move at the same constant speed and update their headings following a nearest-neighbor rule. Velocity cohesion and flocking behavior were observed in both cases. In [6], the nearest-neighbor rule is characterized by a graph-based control law. Moreover, it is also shown that, under some circumstances, the convergence still holds even if the nearest-neighbor set changes over time. The nearest-neighbor rule has the form

$$\dot{x}_i(t) = -k_i \sum_{j \in \mathcal{N}_i} w_{ij}(x_i(t) - x_j(t)), \tag{3}$$

where $x_i(t) \in \mathbb{R}^n$ is the state vector of agent i at time t, and $\mathcal{N}_i$ denotes the neighborhood set of agent i, i.e., the set of agents that interact with agent i. At this stage, we assume that the neighborhood set is time invariant. Later on, $\mathcal{N}_i$ might be time variant when addressing the limited sensor range problem.

Variations of the protocol are mainly different in the ways they evaluate the weight factors k_i and w_{ij}. Linear time-varying weights are used for w_{ij} in [5, 7, 8], where continuous time models are studied. Discrete time models are addressed in [6, 8]. Nonlinear protocols are proposed in [13]. In addition, a robust rendezvous algorithm is presented in [14].

It is worth noting that the convergence of the rendezvous algorithm is, in general, based on the assumption that $\mathcal{N}_i$ is fixed or the underlying graph is connected. However, connection might be lost if connectivity is not considered in the design of the controller. Recently, connectivity has been an issue drawing a great deal of attention. Flocking under switching topologies was studied in [9], where a theoretical frame-work for coordinated control was proposed based on graph theory. In [15], connectedness was preserved by apply nonlinear weights on edges. A measure of the local connectedness of a network was introduced in [16]. The measure depends only on the local communication, which renders the approach a distributed one. In [17], connectivity constraints were related to an individual agent's motion by the construction of a dynamically changing adjacency matrix.

1.2.1.2 Formation Control

The formation control problem is another important problem in networked system control. By a formation, we mean a group of mobile agents moving as a unit while preserving a set of constraints between specific agents. In real applications, the agents live in two- or three-dimensional space and the constraints are normally defined as inter-agent distances. The formation control under a homogeneous network is also

called the leaderless approach [18,19]. Here, the controller is typically given by a mixture of formation-maintenance, obstacle-avoidance, and trajectory-following terms. Furthermore, in [20] it was shown how this approach could be modeled as a hybrid automaton, and in [21], a hybrid control framework was proposed for formation switching. In [5], a method for decentralized information exchange among agents was proposed, which enabled a stable formation. Also, it was shown that a stable information flow combined with a local controller that stabilizes individual agents would result in a stable formation.

Other notable contributions to the general area of networked system control can also be found in the literature. In [22], a set of algorithms was proposed for a multi-robot group to achieve certain geometric formations. In [23, 24], state-dependent dynamic graphs were studied from a combinatoric point-of-view. In [25, 26], swarm stability and cohesion were studied with the aid of social potentials. In [13] artificial potential functions were designed to guarantee stable flocking in both fixed and switching networks. In [27], the complexity of networked systems was studied from an information-theoretic point of view. The so-called algebraic connectivity (or Fiedler value) of a graph is a critical parameter that influences the stability and robustness of a networked system. The problem of maximizing this eigenvalue was studied in [28].

1.2.2 Heterogeneous Network

1.2.2.1 Leader-Follower Structure and Formation Control

In a heterogeneous network, while the majority of the nodes still follows the agreement protocol (3), a small group is not bounded by this protocol and takes the lead. We refer to these advanced agents as *leaders* and the rest as *followers*. We call this kind of network structures, where nodes are divided into a leader set and a follower set, *leader-follower structures*.

A number of formation control approaches are based the leader-follower structure.

In these approaches, either a real agent [29,30] or a virtual agent [31–34] is chosen as the leader, whose movement is constrained by a predefined trajectory. The remaining agents simply track the leader while following some coordination rule to keep the formation. Rigidity becomes one issue of formation keeping and was addressed in [35,36]. Leader-to-formation stability was studied in [37], where the interconnection topology was related to the stability and performance by LFS gains. Furthermore, [37] also suggested that LFS gain could be used to improve the robustness of a formation.

A further extension along this direction is the so-called *containment* problem[38, 39], where the leaders want to keep the followers inside a boundary, like the way dogs herding sheep. In [40], partial difference equations (PdEs) was used over graphs to analyze the behavior of a hybrid leader-follower herding scheme. In [41], a hierarchial containment control strategy is proposed for multi-layer leader-follower structure. Both [40] and [41] adopted a Stop-Go policy to maintain the boundaries.

The controllability issue has recently attracted attentions. It was [42] that first give a necessary and sufficient condition by combing the tools from control theory and algebraic graph theory. In pursuit of a more intrinsic graph theoretical explanation of the controllability in a leader-follower structure, or so-called anchored agreement problem, [43] and [44] respectively give a necessary and a sufficient condition for a leader-follower to be controllable. In their later work [45], they pushed the boundary further by providing a less conservative necessary based on equitable partitions over the graph. A somehow related phenomenon is found in [46], when add a link between two agents maybe "detrimental" to the system performance.

1.2.2.2 Distributed Sensor Networks

Distributed sensor networks (DSNs) have found a number of potential applications, ranging from field surveillance, environment monitoring to geo-scientific exploration [47]. A DSN typically consists of a collection of sensor nodes, each integrated with a microprocessor, a transceiver, and, in some case actuators. Thus, beyond sensing, the

nodes are capable of simple information processing and short distance communication. Though distributed in nature, the information is required to be reported to one or more base stations for data synthesis and decision making. The existence of such central nodes put DSNs under the banner of heterogeneous network.

A common objective of DSN is to collect environmental information, e.g., temperature, radioactivity [48], motion [49], or localization [50] in the covered area through its network of sensors. An example of DSN is shown in Figure 1. The network is set up to monitor the temperature field shown in (a). The DSN is formed by dozens of wireless sensor nodes, as shown in (c), which can measure the temperature. It worth noticing that this mission is different from the sensor network consensus, where the network is monitoring a homogeneous environment and the major mission of the network is to reach an agreement over the parameters being monitored.

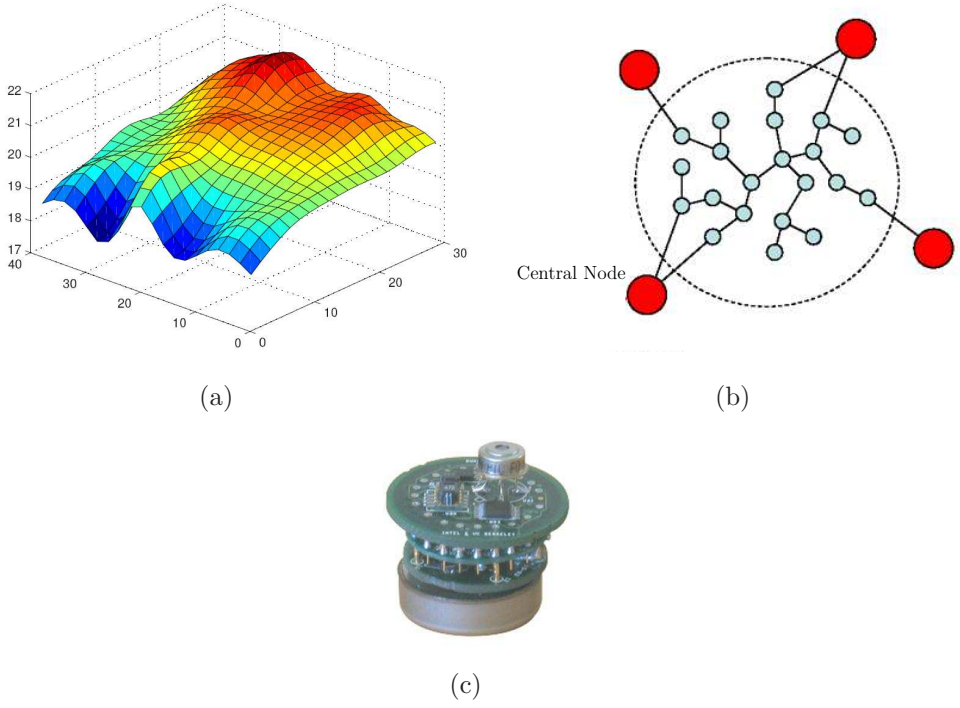


Figure 1. Shown in the pictures are (a) A temperature field, (b) The topology of a distributed sensor network (DSN) with multiple central nodes (big red nodes), and (c) A wireless sensor for temperature measuring.

Sensing with DSNs poses a great challenge to the control and communication society, because it requires the combination of well coordinated sensing, communication and control strategies to overcome the restrictions on power and bandwidth [46]. In a typical DSN, centralized communication, i.e., every node reports its data directly to the central node, is infeasible, because of limited band width. The restricted range of transmission of the sensor nodes, and the fact that they are always spread in a large area, further make a centralized scheme impossible.

Two types of coordination strategies have been proposed in the literature, multi-hop [51] and mobile central nodes [52,53]. In the multi-hop method, the sensor nodes transfer their data to the base station either directly or by ways of other nodes. In such a setting, a intermediate node act as a router. It sends its own data together with the data from its upper link to the next node. Several multi-hop routing protocols are proposed in the literature and the some issues, such as stability, pack losing, are studied. However, the cost of communication, mainly power related, and the error accumulation are two great hurdles for multi-hop method to overcome.

In the mobile central nodes method, the central node(s) can move around and collect data from the nodes that within transmission range. Having avoided the difficulties faced by the multi-hop method, the mobile sensor nodes method has its own limitation. The mobile node can collect information from only a small portion of the network, and the size of the field and its moving speed determine the update frequency. If the area is large and the speed is low, the frequency can be low enough to affect the concurrency of the data.

Other related work can be found in [54] in the context of consensus filter, where the author reduced the distributed Kalman filter into two separate dynamical consensus problem. In [55], authors provided a stochastic strategy to generate optimal sensor trajectories for the coverage problem of mobile sensor network. Estimation problems in a network with packet dropping was studied in [56], where a optimal strategy was

presented. This strategy requires every node in the communication link to store and process the packet they received.

1.3 Organization

This dissertation is organized into two parts. First, we investigated the connectedness issue in homogeneous networks, where the individual agents are similar, is addressed in different context. Notations from algebraic graph theory are reviewed, and graph-based modeling for networked systems is introduced in Chapter 2. Chapter 3 is devoted to the connectedness preserving rendezvous, where a dynamically weighting strategy for the graph-based feedback control is discussed. Following similar methodology, the connectedness preserving problem is solved in the context of formation control in Chapter 4. Then, in Chapter 5.2, the role assignment problem is investigated, which connects the rendezvous and formation control and forms a unified formation process.

In the second part, our focus is switched onto heterogeneous network, where some agents in the network are endowed with greater communication and processing power, and thus take greater responsibilities. Leader-follower structures are networks, where a few advanced agents take leaders role and follows external signal while the rest agents in the network simply follow the leaders. Controllability problem of the leader-follower structure is studied in Chapter 6 and a necessary condition for the system to be controllable is provided. In Chapter 7, a innovative estimation strategy for the distributed sensor network is proposed. Moreover, the observability problem is raised in the context of the estimation strategy we proposed for the distributed sensor network and a necessary condition is given. Containment problem is studied in Chapter 8 and the leader-follower structure is further extended to a hierarchial structure where the complexity issue is also investigated. Several application examples are presented in Chapter 9.

CHAPTER 2

GRAPH-BASED NETWORK MODELS

Graphs are adopted as encodings of the limited information present in networked systems, where edges between nodes correspond to shared information. In this part, we introduce some basic notation and concepts in graph-based modeling and discuss some useful properties related to the matrix representations of graphs.

2.1 Basic Notation and Concepts in Graph Theory

An (undirected) *graph* $\mathcal{G}$ is defined by a set $V = \{1, \dots N\}$ of *nodes* and a set $E \subset V \times V$ of *edges*. A graph with node set V is said to be a graph *on* V, and it can be visually depicted by drawing a dot for each node and a line for each edge, as shown in Figure 2. The number of nodes of a graph $\mathcal{G}$ is its *order*, and its total number of edges is its *degree*. The node set and edge set of a graph $\mathcal{G}$ are referred to as $V(\mathcal{G})$ and $E(\mathcal{G})$, respectively. If we use $|\cdot|$ to denote cardinality, we have that the order of $\mathcal{G}$ is $|V(\mathcal{G})|$ and its degree of $\mathcal{G}$ is $|E(\mathcal{G})|$.

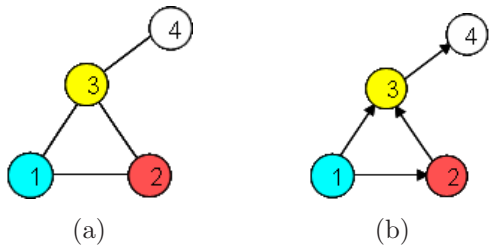


Figure 2. The figure shows (a) a graph on $V = \{1, 2, 3, 4\}$ with edge set $E = \{(1, 2), (1, 3), (2, 3), (3, 4)\}$, and (b) the oriented version of (a).

Two nodes i and j are *adjacent*, or *neighbors*, if $(i, j) \in E(\mathcal{G})$, and the neighboring relation is indicated with $i \sim j$, while $\mathcal{N}_{\mathcal{G}}(i) = \{j \in V(\mathcal{G}) : j \sim i\}$ collects all neighbors to the node i. The *degree* of a node is given by the number of its neighbors, and we say that a graph is *regular* if all nodes have the same degree. If all the nodes of $\mathcal{G}$ are pairwise adjacent, then $\mathcal{G}$ is *complete*. The complete graph with N nodes is

denoted by K_N.

A *path* $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1, \dots, L$, and a graph $\mathcal{G}$ is *connected* if there is a path between any pair of distinct nodes. The number of edges of a path is its *length*.

Let $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ be two graphs. We call $\mathcal{G}'$ a *subgraph* of $\mathcal{G}$ (and $\mathcal{G}$ a *supergraph* of $\mathcal{G}'$) if $V' \subseteq V$ and $E' \subseteq E$, and we denote this by $\mathcal{G}' \subseteq \mathcal{G}$. A subgraph $\mathcal{G}'$ is said to be *induced* from the original graph $\mathcal{G}$ if $E' = E \cap V' \times V'$. In other words, it is obtained by deleting a subset of nodes and all the edges connecting to those nodes. $\mathcal{G}' \subseteq \mathcal{G}$ is a *spanning* subgraph of $\mathcal{G}$ if $V' = V$.

2.2 Matrices Associated with Graphs

The *adjacency* matrix of a graph $\mathcal{A} \in \mathbb{R}^{N \times N}$, with N denoting the number of nodes, is defined by

$$[\,\mathcal{A}(\mathcal{G})\,]_{ij} := \begin{cases} 1 & \text{if} \quad (i,j) \in E(\mathcal{G}) \\ 0 & \text{otherwise.} \end{cases}$$

Given that an orientation has been associated with a graph with M edges, the node-edge *incidence* matrix $\mathcal{B} \in \mathbb{R}^{N \times M}$ is defined as

$$[\,\mathcal{B}(\mathcal{G})\,]_{kl} := \begin{cases} 1 & \text{if} \quad \text{node } k \text{ is the head of edge } l \\ -1 & \text{if} \quad \text{node } k \text{ is the tail of edge } l \\ 0 & \text{otherwise,} \end{cases}$$

where k and l are the indices running over the node and edge sets, respectively.

A matrix that plays a central role in this thesis is the graph *Laplacian* matrix, defined by

$$\mathcal{L}(\mathcal{G}) := \mathcal{B}(\mathcal{G})\,\mathcal{B}(\mathcal{G})^T, \tag{4}$$

from which it follows that the graph Laplacian is a symmetric, positive semi-definite matrix.

As an example, the adjacency matrix of graph (a) in Figure 2 is given by

$$\mathcal{A}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

and the incidence matrix and Laplacian matrix of graph (b) in Figure 2 are given by

$$\mathcal{B}(\mathcal{G}) = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathcal{L}(\mathcal{G}) = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

Let d_i be the degree of node i and let $\mathcal{D}(\mathcal{G}) := \mathbf{Diag}([d_i]_{i=1}^N)$ be the corresponding diagonal degree matrix. It is easy to verify that $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$.

As the Laplacian matrix is positive semi-definite, its spectrum can be ordered as

$$0 = \lambda_1(\mathcal{L}(\mathcal{G})) \leq \lambda_2(\mathcal{L}(\mathcal{G})) \leq \ldots \leq \lambda_N(\mathcal{L}(\mathcal{G})),$$

with $\lambda_i(\mathcal{L}(\mathcal{G}))$ being the i-th ordered eigenvalue of $\mathcal{L}(\mathcal{G})$. It turns out that the multiplicity of the zero eigenvalue of the graph Laplacian is equal to the number of connected components of the graph [57]. In fact, the second smallest eigenvalue $\lambda_2(\mathcal{L}(\mathcal{G}))$ provides a measure of the connectivity of $\mathcal{G}$.

To summarize, the graph Laplacian matrix has a number of well-studied properties [57], including the following:

1. The Laplacian matrix is orientation independent.

2. $\mathcal{L}(\mathcal{G})$ is symmetric and non-negative definite.

3. $\lambda_2(\mathcal{L}(\mathcal{G})) > 0$ if and only if $\mathcal{G}$ is connected.

4. If $\mathcal{G}$ is connected, the eigenvectors $\nu_1, \ldots, \nu_N$ form an orthogonal basis in $\mathbb{R}^N$, and $\nu_1 = 1/\sqrt{N}\mathbf{1}$, where $\mathbf{1}$ denotes the vector with every entry equal to one. In other words, if $\mathcal{G}$ is connected, the null space of $\mathcal{L}(\mathcal{G})$, Null$\mathcal{L}(\mathcal{G})$, is span$\{\mathbf{1}\}$.

If we associate positive weights with the edges, we can define the weighted graph Laplacian matrix as

$$\mathcal{L}_{\mathcal{W}}(\mathcal{G}) := \mathcal{B}(\mathcal{G})\mathcal{W}\mathcal{B}(\mathcal{G})^T, \tag{5}$$

where $\mathcal{W} = \mathbf{Diag}(w_1, \ldots, w_M) \in \mathbb{R}^{M \times M}$, is a diagonal positive definite weight matrix.

2.3 Graph-Based Modeling

Now, we are in the position to model networked systems in terms of graphs. We associate an *interaction graph* with the available information flow, where the nodes correspond to agents, and edges to available, inter-agent communication links. Such interaction graphs are thus representative of the underlying network topology induced by limited information and localized interactions.

Of particular importance to the development in the first part of this work are the Δ-disk proximity graphs, as shown in Figure 3. In a Δ-disk proximity graph, edges are established between nodes i and j if and only if the agents are within distance Δ of each other, i.e., when $\|x_i - x_j\| \leq \Delta$.

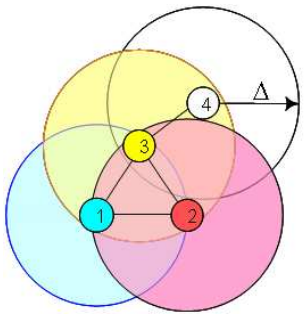


Figure 3. The figure shows a Δ-disk proximity graph on $V = \{1, 2, 3, 4\}$.

It should be noted, already at this point, that such graphs are dynamic in nature

in that edges may appear or disappear as agents move in or out of sensing (or communication) distance of each other. Moreover, it is conceivable that agents are added or removed themselves, making not only the edge set but also the node set a dynamical structure. In this thesis we will not study this latter situation and thus restrict the node set to be static while allowing the edge set to undergo dynamic changes.

CHAPTER 3

GRAPH-BASED CONNECTEDNESS-PRESERVING RENDEZVOUS

3.1 Introduction

In this chapter we will focus on providing solutions to the agreement problem that preserve connectedness in the presence of limited sensing and communication ranges. It should be noted that these problems have already been solved if either connectedness is assumed [4, 7, 13], or connectedness is only required at distinct times [3, 6, 14, 22, 58, 59] in the sense that the agents sense their environment and then moves in such a way that the network is connected at the sensing times, where the agents may be operating synchronously or asynchronously. In particular, the first solution to the connectedness-preserving rendezvous problem was given by Ando et. al. in [3]. There a discrete-time control algorithm was proposed that evaluated and ensured connectivity, as well as other constraints, at each instant of (discrete time). An additional relevant contribution along these lines can be found in [60], where the connectivity-maintenance problem for ad-hoc networks with discrete-time, double integrator dynamics was considered. Different from above solutions, we provide a continuous time solution which guarantees that the graph stay connected for all times during rendezvous.

3.2 Non-weighted Graph-Based Feedback Control

Before we study the dynamic interaction graph, let us revisit some previously established results. Assume, first, that the agents have established communication links between predefined agents and these links are assumed to be available throughout the duration of the maneuver. In other words, the network is modeled as a static interaction graph (SIG) $\mathcal{G}$, where $\mathcal{N}_{\mathcal{G}}(i)$, the neighborhood set of i, is time-invariant.

A more general representation of the nearest neighbor rule in (3) is that

$$u_i = \sum_{j \in \mathcal{N}_\sigma(i)} f(x_i - x_j), \tag{6}$$

where $\mathcal{N}_\sigma(i) \subseteq \mathcal{N}_\mathcal{G}(i)$. The symmetric indicator function $\sigma(i,j) = \sigma(j,i) \in \{0,1\}$ determines whether or not the information available through edge (i,j) should be taken into account, with

$$j \in \mathcal{N}_\sigma(i) \;\Leftrightarrow\; (i,j) \in E(\mathcal{G}) \wedge \sigma(i,j) = 1. \tag{7}$$

(Using the terminology in [61], just because two nodes are "neighbors" it does not follow that they are "friends".) Along the same lines, the decentralized control law $f(x_i - x_j)$ is assumed to be anti-symmetric, i.e.,

$$f(x_i - x_j) = -f(x_j - x_i),\ \forall\ (i,j) \in E(\mathcal{G}). \tag{8}$$

A few remarks about these particular choices of control laws and indicator functions should be made. First of all, the fact that we only allow f to depend on the relative displacements between interacting agents is that this is in general the only type of information available to range-sensor based information channels, where agent i simply measures the position of agent j relative to its current position.

The type of control terms presented in (6) have appeared repeatedly in the multi-agent coordination community, and an intuitive, linear control law for solving the rendezvous problem is given by

$$\begin{aligned} &\sigma(i,j) = 1 \\ &f(x_i - x_j) = -(x_i - x_j) \end{aligned} \qquad \forall\ (i,j) \in E(\mathcal{G}),$$

which gives that

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_\mathcal{G}(i)} (x_i - x_j),\ i = 1, \ldots, N. \tag{9}$$

Under the dynamics in (9), it has been shown that all agents approach the same point asymptotically, provided that the SIG is connected. And, even though this is a

well-established result (see for example [7]), we outline a proof in order to establish some needed notation and tools.

If we now let the n-dimensional position of agent i be given by $x_i = [x_{i,1}, \ldots, x_{i,n}]$, $i = 1, \ldots, N$, and let $x = [x_1^T, \ldots, x_N^T]^T$, we can define the componentwise operator as

$$c(x, j) = [x_{1,j}, \ldots, x_{N,j}]^T \in \mathbb{R}^N, \ j = 1, \ldots, n.$$

Using this notation, together with the observation that (9) can be decoupled along each dimension, we can in fact rewrite (9) as

$$\frac{d}{dt}c(x, j) = -\mathcal{L}(\mathcal{G})c(x, j), \ j = 1, \ldots, n. \tag{10}$$

Now, as pointed out in [7] and [57], if $\mathcal{G}$ is connected then the eigenvector corresponding to the semi-simple eigenvalue 0 is $\mathbf{1}$. This, together with the non-negativity of $\mathcal{L}(\mathcal{G})$ and the fact that span$\{\mathbf{1}\}$ is $\mathcal{L}(\mathcal{G})$-invariant, is sufficient to show that $c(x, j)$ approaches span$\{\mathbf{1}\}$ asymptotically.

This result, elegant in its simplicity, can in fact be extended to dynamic graphs as well. In fact, since $c(x, j)^T c(x, j)$ is a Lyapunov function to the system in (9), for any connected graph $\mathcal{G}$, the control law

$$\frac{d}{dt}c(x(t), j) = -\mathcal{L}(\mathcal{G}(t))c(x(t), j), \tag{11}$$

drives the system to span$\{\mathbf{1}\}$ asymptotically *as long as* $\mathcal{G}(t)$ *is connected for all* $t \geq 0$.

This well-known result is very promising since dynamic network graphs are frequently occurring in that all real sensors and transmitters have finite range. This means that information exchange links may appear or be lost as the agents move around. In fact, if we focus our attention on Δ-disk proximity graphs, we get the Dynamic Interaction Graph (DIG) $\mathcal{G}(t) = (V, E(t))$, where $(i, j) = (j, i) \in E(t)$ if and only if $\|x_i(t) - x_j(t)\| \leq \Delta$.

By applying the control law in (9) to such DIGs, we get a system behavior that seemingly solves the rendezvous problem quite nicely. However, the success of the

control in (9) hinges on an assumption that it shares with most graph-based results, e.g. [6, 13], namely on the connectedness assumption. Unfortunately, this property has to be assumed rather than proved, and in Figure 4 an example is shown where connectedness is lost when using (11) to control a system whose network topology is a Δ-disk proximity DIG.

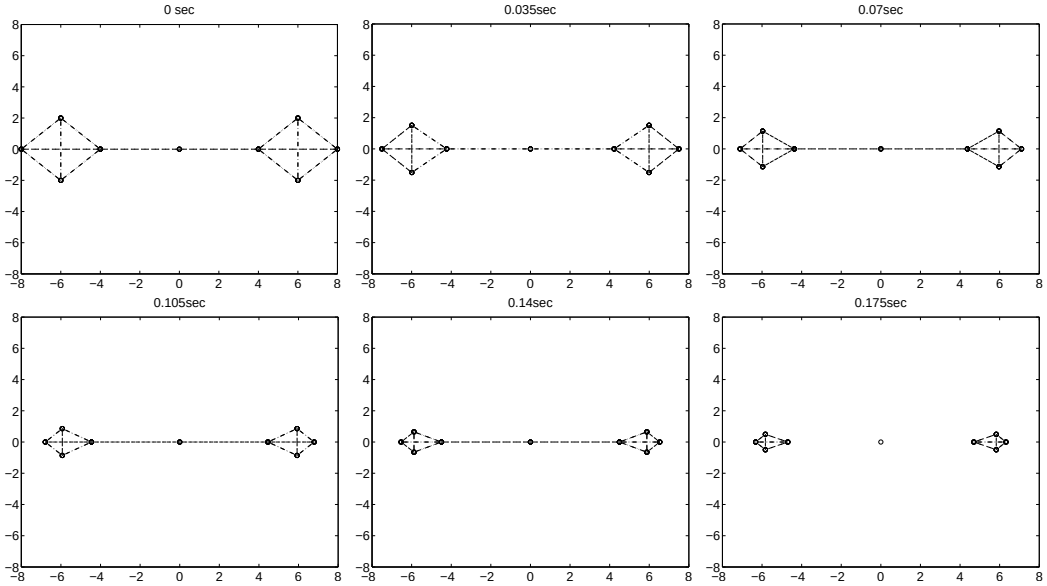


Figure 4. A progression is shown where connectedness is lost even though the initial graph is connected ($\Delta = 4.5$).

What we will do for the remainder of this chapter is to show how this assumption can be overcome by modifying the control law in (9) in such a way that connectedness holds for all times, while ensuring that the control laws are still based solely on local information, in the sense of (6).

3.3 Weighted Graph-Based Feedback Control

3.3.1 Static Graph

In this section, we still restrict the interaction graphs to be static, i.e., we will only study the SIG-case in which the behavior of the multi-agent system is defined through a fixed network topology. However, we will show how the introduction of nonlinear edge-weights can be used to establish certain invariance properties, which will enable

us to extend the graph-based feedback law to a dynamic interaction graph case.

To arrive at the desired invariance properties, we will first investigate decentralized control laws. For this, we need to establish some additional notation. Given an edge $(i,j) \in E(\mathcal{G})$, we let $\ell_{ij}(x)$ denote the edge vector between the agents i and j, i.e., $\ell_{ij}(x) = x_i - x_j$. Next, we assign a strictly positive and bounded weight w to each edge in the SIG, based solely on $\ell_{ij}(x)$. Specifically, let $w_{ij}(x) = w(\ell_{ij}(x))$, where $w : \mathbb{R}^n \to \mathbb{R}_+$ is a positive, symmetric weight function.

The control law that we are interested in is of the form

$$\begin{aligned} &\sigma(i,j) = 1 \\ &f(x_i - x_j) = -w_{ij}(x)(x_i - x_j) \end{aligned} \qquad \forall\ (i,j) \in E(\mathcal{G}), \tag{12}$$

The problem of how to maintain connectedness boils down to the problem of how to choose the weight w.

This choice of decentralized control law gives

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_{\mathcal{G}}(i)} w_{ij}(x)(x_i - x_j), \tag{13}$$

which can be rewritten as

$$\frac{d}{dt} c(x,j) = -\mathcal{B}\mathcal{W}(x)\mathcal{B}^T c(x,j),\ j = 1, \ldots, n, \tag{14}$$

where $\mathcal{W}(x)$ =diag$(w_1(x), \ldots, w_M(x)) \in \mathbb{R}^{M \times M}$, where, as before, $M = |E(\mathcal{G})|$) is the total number of edges, and where we have associated a label in $\{1, \ldots, M\}$ with each of the edges.

We can then define the state-dependent, weighted graph Laplacian as

$$\mathcal{L}_{\mathcal{W}}(x) = \mathcal{B}\mathcal{W}(x)\mathcal{B}^T, \tag{15}$$

where, as before, $\mathcal{W}(x) \in \mathbb{R}^{M \times M}$ is a diagonal matrix with each element corresponding to a strictly positive edge weight. It is moreover straightforward to establish that as long as the graph is connected, $\mathcal{L}_{\mathcal{W}}(x)$ is still positive semidefinite, with only one zero eigenvalue corresponding to the null-space span$\{\mathbf{1}\}$.

What we would like to show is that, given a critical distance δ, together with appropriate edge-weights, the edge-lengths never go beyond δ if they start out being less than $\delta - \epsilon$, for some arbitrarily small $\epsilon \in (0, \delta)$.

We moreover define the ϵ-interior of a δ-constrained realization of a SIG, $\mathcal{G}$, as

$$\mathcal{D}^{\epsilon}_{\mathcal{G},\delta} = \{x \in \mathbb{R}^{nN} \mid \|\ell_{ij}\| \leq (\delta - \epsilon),\ \forall (i,j) \in E(\mathcal{G})\},$$

where $x = [x_1^T, \ldots, x_N^T]^T$ is the collection of the states of all the agents. $\mathcal{D}^{\epsilon}_{\mathcal{G},\delta}$ denote all the network realizations of graph $\mathcal{G}$ such that the distance along each edge is shorter than $\delta - \epsilon$.

An edge-tension function $\mathcal{V}_{ij}$, can then be defined as

$$\mathcal{V}_{ij}(\delta, x) = \begin{cases} \frac{\|\ell_{ij}(x)\|^2}{\delta - \|\ell_{ij}(x)\|} & \text{if } (i,j) \in E(\mathcal{G}) \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

with

$$\frac{\partial \mathcal{V}_{ij}(\delta, x)}{\partial x_i} = \begin{cases} \frac{2\delta - \|\ell_{ij}(x)\|}{(\delta - \|\ell_{ij}(x)\|)^2}(x_i - x_j) & \text{if } (i,j) \in E(\mathcal{G}) \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Note that this edge-tension function (as well as its derivatives) is infinite when $\|\ell_{ij}(x)\| = \delta$ for some i, j, and, as such, it may seem like an odd choice. However, as we will see, we will actually be able to prevent the energy to reach infinity, and instead we will study its behavior on a compact set on which it is continuously differentiable.

The total tension energy of $\mathcal{G}$ can now be defined as

$$\mathcal{V}(\delta, x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{V}_{ij}(\delta, x). \tag{18}$$

Lemma 3.3.1. *Given an initial position $x_0 \in \mathcal{D}^{\epsilon}_{\mathcal{G},\delta}$, for a given $\epsilon \in (0, \delta)$. If the SIG $\mathcal{G}$ is connected then the set $\Omega(\delta, x_0) := \{x \mid \mathcal{V}(\delta, x) \leq \mathcal{V}(\delta, x_0)\}$ is an invariant set to the system under the control law*

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_{\mathcal{G}}(i)} \frac{2\delta - \|\ell_{ij}(x)\|}{(\delta - \|\ell_{ij}(x)\|)^2}(x_i - x_j). \tag{19}$$

Proof. We first note that the control law in (19) can be rewritten as

$$\dot{x}_i = -\sum_{j\in\mathcal{N}_{\mathcal{G}}(i)} \frac{\partial \mathcal{V}_{ij}(\delta,x)}{\partial x_i} = -\frac{\partial \mathcal{V}(\delta,x)}{\partial x_i} = -\nabla_{x_i}\mathcal{V}(\delta,x).$$

This expression may be ill-defined since it is conceivable that the edge-lengths approach δ and what will be shown is that this will not happen. In fact, assume that at time τ we have $x(\tau) \in \mathcal{D}^{\epsilon'}_{\mathcal{G},\delta}$ for some $\epsilon' > 0$. Then the time derivative of $\mathcal{V}(\delta, x(\tau))$ is

$$\begin{aligned}\dot{\mathcal{V}}(\delta,x(\tau)) &= \nabla_x \mathcal{V}(\delta,x(\tau))^T \dot{x}(\tau)\\ &= -\sum_{i=1}^{N} \dot{x}_i(\tau)^T \dot{x}_i(\tau)\\ &= -\sum_{j=1}^{n} c(x(\tau),j)^T \mathcal{L}_{\mathcal{W}}(\delta,x(\tau))^2 c(x(\tau),j),\end{aligned} \tag{20}$$

where $\mathcal{L}_{\mathcal{W}}(\delta,x)$ is given in (15), with weight positive definite (on $\Omega(\delta,x_0)$) matrix $\mathcal{W}(\delta,x)$

$$\begin{aligned}\mathcal{W}(\delta,x) &= \mathrm{diag}(w_k(\delta,x)), \quad k = 1,2,\ldots,M,\\ w_k(\delta,x) &= \frac{2\delta - \|\ell_k(x)\|}{(\delta - \|\ell_k(x)\|)^2},\end{aligned} \tag{21}$$

where we have arranged the edges such that subscript k corresponds to edge k. We will use this notation interchangeably with w_{ij} and ℓ_{ij}, whenever it is clear from the context.

Note that for any ϵ' bounded away from 0 from below and δ from above, and for any $x \in \mathcal{D}^{\epsilon'}_{\mathcal{G},\delta}$, the time derivative of the total tension energy is well-defined. Moreover, for any such x, $\mathcal{V}(\delta,x)$ is non-negative and $\dot{\mathcal{V}}(\delta,x)$ is non-positive (since $\mathcal{L}_{\mathcal{W}}(\delta,x)$ is positive semidefinite for all $x \in \Omega(\delta,x_0)$). Hence, in order to establish the invariance of $\Omega(\delta,x_0)$, all that needs to be shown is that, as $\mathcal{V}$ decreases (or at lest does not increase), no edge-distances will tend to δ. In fact, since $\mathcal{D}^{\epsilon}_{\mathcal{G},\delta} \subset \mathcal{D}^{\epsilon'}_{\mathcal{G},\delta}$ if $\epsilon > \epsilon'$, we would have established the invariance of $\Omega(\delta,x_0)$ if we could find an $\epsilon' > 0$ such that, whenever the system starts from $x_0 \in \mathcal{D}^{\epsilon}_{\mathcal{G},\delta}$, we can ensure that it never leaves the superset $\mathcal{D}^{\epsilon'}_{\mathcal{G},\delta}$.

Let

$$\hat{\mathcal{V}}_\epsilon := \max_{x \in \mathcal{D}^\epsilon_{\mathcal{G},\delta}} \mathcal{V}(\delta, x).$$

This maximum always exists and is obtained when all edges are at the maximal allowed distance $\delta - \epsilon$, i.e.,

$$\hat{\mathcal{V}}_\epsilon = \frac{M(\delta - \epsilon)^2}{\epsilon},$$

which is a monotonously decreasing function in ϵ over $(0, \delta)$.

What we will show next is that we can bound the maximal edge distance that can generate this total tension energy, and the maximal edge-length $\hat{\ell}_\epsilon \geq \delta - \epsilon$ is one where the entire total energy is contributed from that one single edge. In other words, all other edges have length 0, and the maximal edge length satisfies

$$\hat{\mathcal{V}}_\epsilon = \frac{\hat{\ell}_\epsilon^2}{\delta - \hat{\ell}_\epsilon},$$

i.e.,

$$\frac{M(\delta - \epsilon)^2}{\epsilon} = \frac{\ell_\epsilon^2}{\delta - \ell_\epsilon},$$

which implies that

$$\hat{\ell}_\epsilon \leq \delta - \frac{\epsilon}{M} < \delta.$$

Hence ℓ_ϵ is bounded away from above from δ and it is moreover bounded from above by a strictly decreasing function in ϵ on $(0, \delta)$. Hence, as $\mathcal{V}$ decreases (or at least is non-increasing), no edge-distances will tend to δ, which completes the proof. ∎

The invariance of $\Omega(\delta, x_0)$ now leads us to the main SIG theorem.

Theorem 3.3.2. *Given a connected SIG $\mathcal{G}$ with initial condition $x_0 \in \mathcal{D}^\epsilon_{\mathcal{G},\delta}$, for a given $\epsilon > 0$. Then the the multi-agent system under the control law in (19) asymptotically converges to the static centroid $\bar{x}(x_0)$.*

Proof. The proof of convergence is based on LaSalle's invariance theorem. Let $\mathcal{D}^\epsilon_{\mathcal{G},\delta}$ and $\Omega(\delta, x_0)$ be defined as before. From Lemma 3.3.1, we know that $\Omega(\delta, x_0)$ is

positively invariant with respect to the dynamics in (19). We also note that span$\{\mathbf{1}\}$ is $\mathcal{L}_{\mathcal{W}}(\delta, x)$-invariant for all $x \in \Omega(\delta, x_0)$. Hence, because of the fact that $\dot{\mathcal{V}}(\delta, x) \leq 0$, with equality only when $c(x(t), j) \in \text{span}\{\mathbf{1}\},\ \forall j \in \{1, \ldots, n\}$, convergence to span$\{\mathbf{1}\}$ follows.

Next we need to show that the agents converge to the centroid. The centroid is given by

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i,$$

and the component-wise dynamics of the centroid is

$$\frac{d}{dt}\overline{c(x,j)} = \frac{1}{N}\mathbf{1}^T \frac{d}{dt} c(x,j) = -\frac{1}{N}\mathbf{1}^T \mathcal{L}_{\mathcal{W}}(\delta, x) c(x,j).$$

Now, since $\mathbf{1}^T \mathcal{L}_{\mathcal{W}}(\delta, x) = (\mathcal{L}_{\mathcal{W}}(\delta, x)\mathbf{1})^T = 0,\ \forall x \in \Omega(x_0)$, we directly have that $\dot{\bar{x}} = 0$, i.e., the centroid is static, determined entirely by the initial condition x_0. As such, we can denote the centroid by $\bar{x}(x_0)$. (This is in fact just a special case of the observation that the centroid is static under any control law in (6).)

Now, let $\bar{\xi} \in \mathbb{R}^N$ be any point on span$\{\mathbf{1}\}$ (i.e., $\bar{\xi} = (\xi, \ldots, \xi)^T$ for some $\xi \in \mathbb{R}$) that is consistent with a static centroid. This implies that

$$\overline{c(x,j)} = \frac{1}{N} \sum_{i=1}^{N} \xi = \xi,$$

and hence ξ has to be equal to the centroid itself. As a consequence, if $x_i,\ i = 1, \ldots, n$, converged anywhere other than the centroid, we would have a contradiction, and the proof follows. ∎

Note that the construction we have described corresponds to adding nonlinear, state-dependent weights to the edges in the graph. One could conceivably also add weights to the nodes as well. Unless these weights were all equal, they would violate the general assumption in (6), but for the sake of completeness, we briefly discuss this situation in the next few paragraphs.

A node weight would be encoded in the dynamics of the system through the weight matrix $D(x)$ as

$$\frac{dc(x,j)}{dt} = -D(x)\mathcal{L}_{\mathcal{W}}(x)c(x,j),\ j = 1,\dots,n.$$

As long as $D(x)$ is diagonal and positive definite for all x, (with the diagonal elements bounded away from 0), the null-space remains $\text{null}(D(x)\mathcal{L}_{\mathcal{W}}(x)) = \text{span}\{\mathbf{1}\},\ \forall x \in \mathbb{R}^{nN}$, and the controller still drives the system to $\text{span}\{\mathbf{1}\}$. However, it is straightforward to show that in this case the positions $x_i \in \mathbb{R}^n,\ i = 1,\dots,N$, approach the same static point $\bar{x}_D(x_0) \in \mathbb{R}^n$, given by

$$\bar{x}_D(x_0) = \frac{1}{\text{tr}(D^{-1}(x_0))}\sum_{i=1}^{N}(d_i^{-1}(x_0))x_{0,i}, \tag{22}$$

where $x_{0,i} \in \mathbb{R}^n,\ i = 1,\dots,N$, is the initial location of agent i, $d_i(x)$ is the ith diagonal element of $D(x)$, and $\text{tr}(D(x))$ denotes the trace of matrix $D(x)$.

That concludes this section, where a static information topology, or SIG, was assumed. In what follows, we will show that a similar strategy can be employed even if the graph is allowed to change as the agents move around in the environment.

3.3.2 Dynamic Graphs

As mentioned before, during a maneuver, the interaction graph $\mathcal{G}$ may change as the different agents move in and out of each others sensory ranges. What we focus on in this section is whether or not an argument, similar to the previous stability result, can be constructed for the case when $(i,j) \in E(\mathcal{G})$ if and only if $\|x_i - x_j\| \leq \Delta$.

In fact, we intend to reuse the tension energy from the previous section, with the particular choice of $\delta = \Delta$. However, since in (21)

$$\lim_{\|\ell_k\|\uparrow\Delta} w_k(\Delta, \|\ell_k\|) = \infty,$$

we can not directly let the inter-agent tension energy affect the dynamics as soon as two agents form edges in between them, i.e., as they move within distance Δ of each other. The reason for this is that we can not allow infinite tension energies in

the definition of the control laws. To overcome this problem, we chose to introduce a certain degree of hysteresis into the system, through the indicator function σ. In particular, we let $\sigma(i,j)$ be given by the state machine in Figure 5.

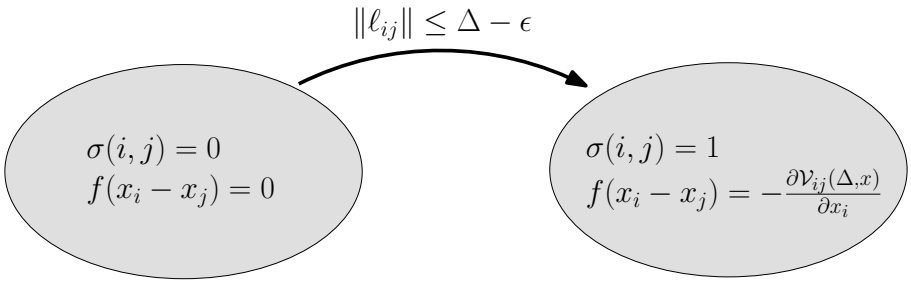


Figure 5. The figure depicts the hysteresis protocol for adding inter-agent tension functions to the total tension function only when agents get within a distance $\Delta - \epsilon$ of each other, rather than when they first encouter each other at a distance Δ.

To elaborate further on the state machine in Figure 5, we let the total tension energy be affected by an edge (i,j) that was previously not contributing to the total energy, when $\|\ell_{ij}\| \leq (\Delta - \epsilon)$, where $\epsilon > 0$ is the predefined *switching threshold*. Once the edge is allowed to contribute to the total tension energy, it will keep doing so for all subsequent times. Note that the switching threshold can take on any arbitrary value in $(0, \Delta)$. The interpretation is simply that a smaller ϵ-value corresponds to a faster inclusion of the inter-robot information into the decentralized control law.

In other words, what we propose for the Δ-disk proximity DIGs is thus to let

$$\begin{aligned} \sigma(i,j)[t^+] &= \begin{cases} 0 & \text{if } \sigma(i,j)[t^-] = 0 \ \wedge \ \|\ell_{ij}\| > \Delta - \epsilon \\ 1 & \text{otherwise} \end{cases} \\ f(x_i - x_j) &= \begin{cases} 0 & \text{if } \sigma(i,j) = 0 \\ -\frac{\partial \mathcal{V}_{ij}(\Delta,x)}{\partial x_i} & \text{otherwise,} \end{cases} \end{aligned} \tag{23}$$

where we have used the notation $\sigma(i,j)[t^+]$ and $\sigma(i,j)[t^-]$ to denote $\sigma(i,j)$'s value before and after the state transition in Figure 5. It is worth noticing that if $\sigma(i,j)[t_0] = 1$ for some t_0, then $\sigma(i,j)[t_0] = 1$ for all $t > t_0$.

Before we can state the rendezvous theorem for dynamic graphs, we also need to

introduce the subgraph $\mathcal{G}_\sigma \subset \mathcal{G}$, induced by the indicator function σ:

$$\mathcal{G}_\sigma = (V(\mathcal{G}), E(\mathcal{G}_\sigma)),$$

where

$$E(\mathcal{G}_\sigma) = \{(i,j) \in E(\mathcal{G}) \mid \sigma(i,j) = 1\}.$$

Theorem 3.3.3. *Given an initial position $x_0 \in \mathcal{D}^{\epsilon}_{\mathcal{G}^0,\Delta}$, where $\epsilon > 0$ is the switching threshold in 23), and where $\mathcal{G}^0$ is the initial Δ-disk DIG. Assume that the graph $\mathcal{G}^0_\sigma$ is connected, where $\mathcal{G}^0_\sigma$ is the graph induced by the initial indicator function value. Then, by using the control law*

$$u_i = -\sum_{j \in \mathcal{N}_\sigma(i)} \frac{\partial \mathcal{V}_{ij}(\Delta, x)}{\partial x_i}, \tag{24}$$

where $\sigma(i,j)$ is given in (23), the group of agents asymptotically converges to span$\{\mathbf{1}\}$.

Proof. Since, from Lemma 3.3.1, we know that no edges in $\mathcal{G}^0_\sigma$ will be lost, only two possibilities remains, namely that no new edges will be added to the graph during the maneuver, or new edges will in fact be added. If no edges are added, then we know from Theorem 3.3.2 that the system will converge to span$\{\mathbf{1}\}$ asymptotically. However, the only graph consistent with $x \in$ span$\{\mathbf{1}\}$ is $\mathcal{G}^0_\sigma = K_N$ (the complete graph over N nodes), and hence no new edges will be added only if the initial, indicator induced graph is complete. If it is not complete, at least one new edge will be added. But, since $\mathcal{G}^0_\sigma$ is an arbitrary connected graph, and connectivity can never be lost by adding new edges, we get that new edges will be added until the indicator induced graph is complete, at which point the system converges asymptotically to span$\{\mathbf{1}\}$. ∎

3.4 Examples

In this section we will show some simulation results that illustrate the proposed coordination control strategies for different problems. In all of these simulations, the cut-off distance for inter-agent sensing and communication is $\Delta = 4$, and the

switching threshold dictating when to add edges is $\epsilon = 0.05$, i.e., a new edge is added only when the corresponding inter-agent distance is $\Delta - \epsilon$.

We first simulate the rendezvous behavior under the weighted Laplacian control law given in (24). In fact, Figure 6 shows the movement of the collection of agents, under exactly the same initial position as in Figure 4. What is different here is, as could be expected, that no links are broken.

In contrast to the first one, Figure 7 depicts the same situation under the addition of a vertex-weight matrix $D = \text{diag}(\{I_4,\ 0.5I_4\})$ to the Laplacian control law, rendering the initial centroid to no longer be static. The trajectories of the different agents are shown in Figure 8, where we, as should be expected, find the convergence to a weighted centroid, as per (22). Because the weights are symmetrically distributed about the y-axis, the rendezvous point is still on the x-axis, yet shifted towards positive x-values, where the agents are more heavily weighted.

3.5 Conclusions

A graph-based nonlinear feedback control law is studied for distributed coordination control of multi agent system. The nonlinear feedback law is based on weighted graph Laplacians and it is proved to solve the rendezvous problem. Furthermore, the proposed control law is also proved to be able to guarantee that the connectedness is not lost during maneuvers.

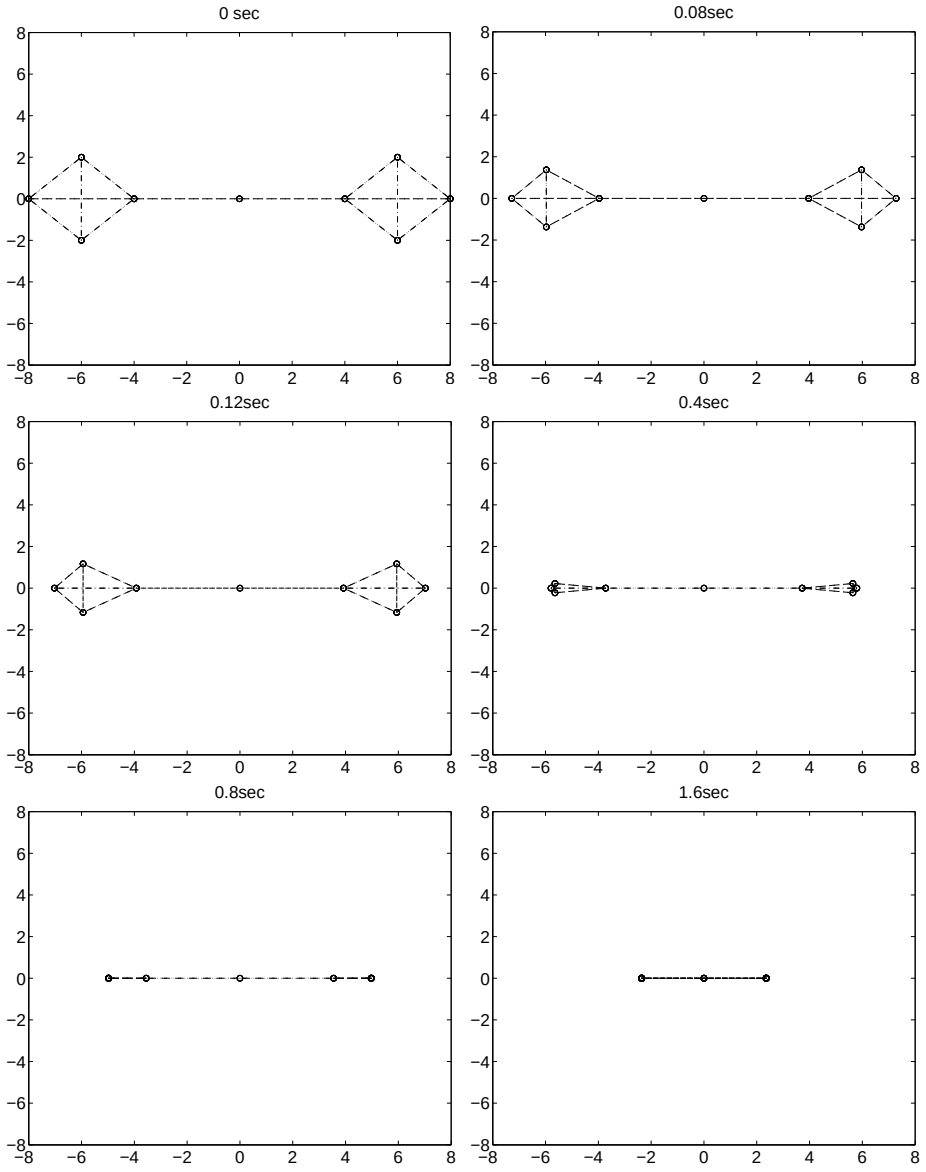


Figure 6. A progression is shown where connectedness is maintained during the rendezvous maneuver, with $D = I$. Depicted are the positions of the agents and the edges in the DIG as a function of time.

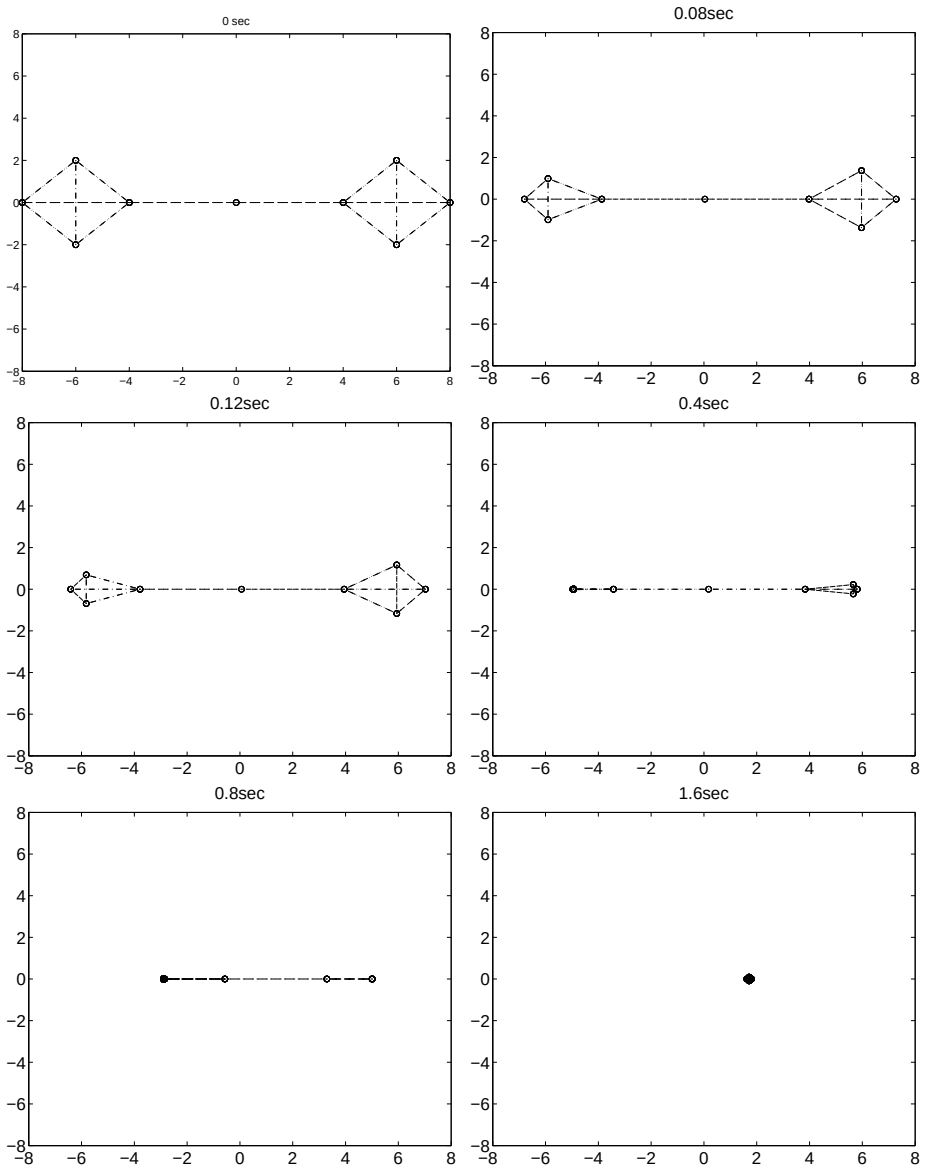
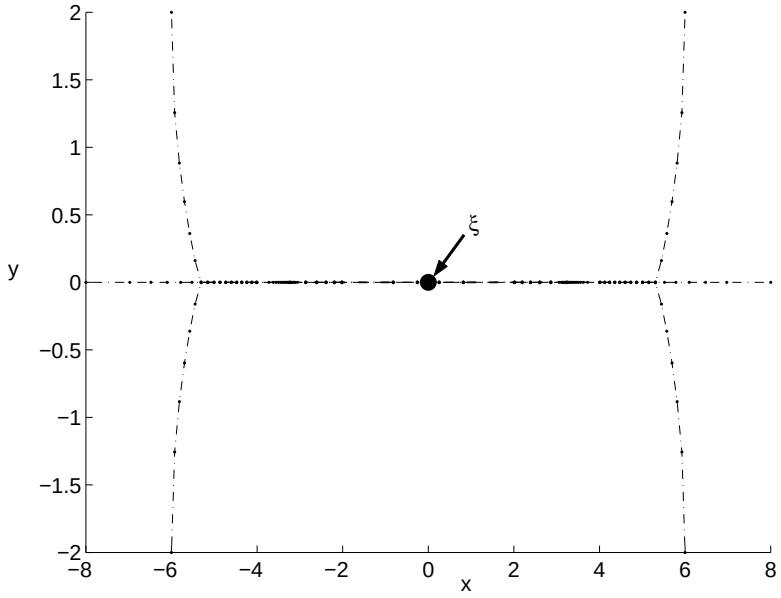
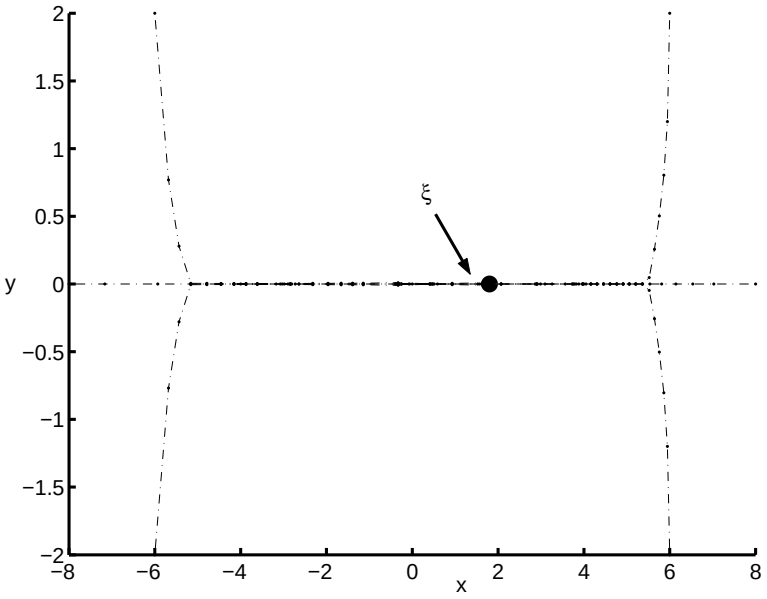


Figure 7. A progression is shown where connectedness is maintained during the rendezvous maneuver, with $D = \mathbf{diag}(\{I_4,\ 0.5I_4\})$.



(a)



(b)

Figure 8. Trajectory for (a) non-weighted and (b) weighted rendezvous.

CHAPTER 4

GRAPH-BASED CONNECTEDNESS-PRESERVING FORMATION CONTROL

4.1 Introduction

In Chapter 3, the connectedness-preserving control method solves the rendezvous problem. In this chapter, we will follow the same methodology to solve the distributed formation control problem. By formation we mean a group of mobile agents preserving a set of constraints among them. By formation control, we understand the problem of driving the collection of mobile agents to some translationally invariant target geometry, i.e., the control objective is to drive the collection of autonomous mobile agents to a specific configuration such that their relative positions satisfy some desired topological and physical constraints. In what follows, we first give a graph model for formations in Chapter 4.2, and propose our graph based control method for formation control, which preserves connectedness, in Section 4.3. In Section 4.4, a hybrid strategy is proposed which can realize formations from arbitrary initial positions. Examples are shown in Section 4.5 and conclusion is drawn in 4.6.

4.2 Graph Model of Formations

The constraints used to specify a formation can be described by a connected, edge-labeled graph $\mathcal{G}_d = (V, E_d, d)$, where the subscript d denotes "desired". Here, E_d encodes the desired robot inter-connections, i.e., whether or not a desired inter-agent distance is specified between two agents or not, and the edge-labels $d : E_d \to \mathbb{R}^n$ defines the desired relative inter-agent displacements, with $\|d_{ij}\| < \Delta$ for all i, j such that $(i, j) \in E_d$. In other words, what we would like is that $x_i - x_j \to d_{ij}$ $\forall i, j$ such that $(i, j) \in E_d$.

One may notice that it is possible that the assignment of general edge-labels

to a DIG may result in conflicting constraints. This is addressed in [62] as the realization problem of connectivity graphs. We will not discuss this problem here and simply assume that the constraints are compatible. Another issue concerning the target formation is that of rigidity, which has been discussed in [10, 35] and [36], and will neither be discussed further in this thesis. Instead, we assume that the target formation is chosen in such a way that rigidity is obtained if, in fact, this is a desired characteristic of the target formation graph-topology.

Given a desired formation, the goal of the distributed formation control is to find a feedback law such that:

F1) The dynamic interaction graph $\mathcal{G}(t)$ converges to a graph that is a supergraph of the desired graph $\mathcal{G}_d$ (without labels) in finite time. In other words, what we want is that $E_d \subset E(t)$ for all $t \geq T$, for some finite $T \geq 0$;

F2) $\|\ell_{ij}(t)\| = \|x_i(t) - x_j(t)\|$ converges asymptotically to $\|d_{ij}\|$ for all i, j such that $(i, j) \in E_d$; and

F3) The feedback law utilizes only local information.

Here "F" stands for "formation" and what will be established is in fact that these properties hold for a particular choice of decentralized control law.

4.3 Graph-Based Formation Control

Analogous to the treatment of the rendezvous problem, we first propose a solution to the formation control problem, and then show that this solution does in fact preserve connectedness as well as guarantee convergence in the sense of F1 and F2 above. The solution will be based on a variation of the previously derived rendezvous controller. In fact, assume that we have established a set of arbitrary targets $\tau_i \in \mathbb{R}^n$ that are consistent with the desired inter-agent displacement, i.e.,

$$d_{ij} = \tau_i - \tau_j, \ \forall \ i, j \text{ s.t. } (i, j) \in E_d.$$

We can then define the displacement from τ_i at time t as

$$y_i(t) = x_i(t) - \tau_i.$$

As before, we let $\ell_{ij}(t) = x_i(t) - x_j(t)$ and we moreover let $\lambda_{ij}(t) = y_i(t) - y_j(t)$, implying that

$$\lambda_{ij}(t) = \ell_{ij}(t) - d_{ij}.$$

Now, under the assumption that $\mathcal{G}_d$ is a connected *spanning graph* of the initial interaction graph $\mathcal{G}$, i.e., $V(\mathcal{G}_d) = V(\mathcal{G})$ and $E_d \subseteq E(\mathcal{G})$, we propose the following control law:

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_{\mathcal{G}_d}(i)} \frac{2(\Delta - \|d_{ij}\|) - \|\ell_{ij} - d_{ij}\|}{(\Delta - \|d_{ij}\| - \|\ell_{ij} - d_{ij}\|)^2}(x_i - x_j - d_{ij}). \tag{25}$$

The reason why this seemingly odd choice makes sense is because we can again use the edge-tension function $\mathcal{V}$ to describe this control law. In particular, using the following parameters in the edge-tension function

$$\mathcal{V}_{ij}(\Delta - \|d_{ij}\|, y) = \begin{cases} \frac{\|\lambda_{ij}\|^2}{\Delta - \|d_{ij}\| - \|\lambda_{ij}\|} & \text{if } (i,j) \in E_d \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

we obtain the decentralized control law

$$\begin{aligned} &\sigma(i,j) = 1 \\ &f(x_i - x_j) = -\tfrac{\partial \mathcal{V}_{ij}(\Delta - \|d_{ij}\|, y)}{\partial y_i} \end{aligned} \qquad \forall\ (i,j) \in E_d. \tag{27}$$

Note that this control law in fact implies something stronger than just measurements of displacement. Instead the agents must also share a common coordinate system. However, they do not need to know their exact location in this coordinate system.

Now, along each individual dimension, the dynamics in (27) becomes

$$\frac{dc(x,j)}{dt} = \frac{dc(y,j)}{dt} = -\mathcal{L}_W(\Delta - \|d\|, y)c(y,j), \quad j = 1, 2, \ldots n,$$

where $\mathcal{L}_W(\Delta - \|d\|, y)$ is the graph Laplacian associated with $\mathcal{G}_d$, weighted by $W(\Delta - \|d\|, y)$, and where we have used the convention that the term $\Delta - \|d\|$ should be interpreted in the following manner:

$$\begin{aligned} \mathcal{W}(\Delta - \|d\|, y) &= \mathrm{diag}(w_k(\Delta - \|d_k\|, y)), \quad k = 1, 2, \dots |E_d|, \\ w_k(\Delta - \|d_k\|, y) &= \frac{2(\Delta - \|d_k\|) - \|\lambda_k\|}{(\Delta - \|d_k\| - \|\lambda_k\|)^2}. \end{aligned} \tag{28}$$

Here, again, the index k runs over the edge set E_d. Note that this construction allows us to study the evolution of y_i, rather than $x_i,\ i = 1, \dots, N$, and we formalize this in the following lemma for static interaction graphs:

Corollary 4.3.1. *Let the total tension energy function be*

$$\mathcal{V}(\Delta - \|d\|, y) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{V}_{ij}(\Delta - \|d_{ij}\|, y). \tag{29}$$

Given $y_0 \in \mathcal{D}^{\epsilon}_{\mathcal{G}_d, \Delta - \|d\|}$, with $\mathcal{G}_d$ being a connected spanning graph, then the set $\Omega(\Delta - \|d\|, y_0) := \{y \mid \mathcal{V}(\Delta - \|d\|, y) \leq \mathcal{V}_0\}$, where $\mathcal{V}_0$ denotes the initial value of the total tension energy function, is an invariant set under the control law in (25), under the assumption that the interaction graph is static.

Proof. By the proposed control law in (25),

$$\dot{y}_i = -\sum_{j \in \mathcal{N}_{\mathcal{G}_d}(i)} \frac{\partial \mathcal{V}_{ij}(\Delta - \|d_{ij}\|, y)}{\partial y_i} = -\frac{\partial \mathcal{V}(\Delta - \|d\|, y)}{\partial y_i} = -\nabla_{y_i} \mathcal{V}(\Delta - \|d\|, y).$$

The non-positivity of $\dot{\mathcal{V}}$ now follows the same argument as in (20) in the proof of Lemma 3.3.1. Moreover, for each initial $y_0 \in \mathcal{D}^{\epsilon}_{\mathcal{G}_d, \Delta - \|d\|}$, the corresponding maximal, total tension-energy induces a maximal possible edge length. Following the same line of reasoning as in the proof of Lemma 3.3.1, the invariance of $\Omega(\Delta - \|d\|, y_0)$ thus follows. ∎

Note that Lemma 4.3.1 says that if we could use $\mathcal{G}_d$ as a SIG, $\Omega(\Delta - \|d\|, y_0)$ is an invariant set. In fact, it is straightforward to show that if $\mathcal{G}_d$ is a spanning graph to the initial proximity Δ-disk DIG, then it remains a spanning graph to $\mathcal{G}(x(t))\ \forall t \geq 0$.

Corollary 4.3.2. *Given an initial condition x_0 such that $y_0 = (x_0 - \tau_0) \in \mathcal{D}^{\epsilon}_{\mathcal{G}_d, \Delta - \|d\|}$, with $\mathcal{G}_d$ being a connected spanning graph of $\mathcal{G}(x_0)$, the group of autonomous mobile agents adopting the decentralized control law in (25) can guarantee that $\|x_i(t) - x_j(t)\| = \|\ell_{ij}(t)\| < \Delta,\ \forall t > 0$ and $(i,j) \in E_d$.*

Proof. Given two agents i, j that are adjacent in $\mathcal{G}_d$, and suppose that $\|\lambda_{ij}\| = \|y_i - y_j\|$ approaches $\Delta - \|d_{ij}\|$. Since $\mathcal{V}_{ij} \geq 0,\ \forall i, j$ and $t > 0$, as well as

$$\lim_{\|\lambda_{ij}\| \uparrow (\Delta - \|d_{ij}\|)} \mathcal{V}_{ij} = \infty,$$

this would imply that $\mathcal{V} \to \infty$, which contradicts Lemma 4.3.1. As a consequence, $\|\lambda_{ij}\|$ is bounded away from $\Delta - \|d_{ij}\|$. This means that

$$\|\ell_{ij}\| = \|\lambda_{ij} + d_{ij}\| \leq \|\lambda_{ij}\| + \|d_{ij}\| < \Delta - \|d_{ij}\| + \|d_{ij}\| = \Delta,$$

and hence edges in E_d are never lost under the control law in (25). In other words, $\|l_{ij}(t)\| < \Delta,\ \forall t \geq 0$, which in turn implies that connectedness is preserved. ∎

We have thus established that if $\mathcal{G}_d$ is a spanning graph of $\mathcal{G}(x_0)$ then it remains a spanning graph of $\mathcal{G}(x(t)),\ \forall t > 0$ (under certain assumptions on x_0), even if $\mathcal{G}(x(t))$ is given by a Δ-disk DIG. And, since the control law in (25) only takes pairwise interactions in E_d into account, we can view this dynamic situation as a static situation, with the SIG being given by $\mathcal{G}_d$. What remains to be shown is that the system in fact converges in the sense of the formation control properties F1, F2, and F3, as previously defined. That F3 (decentralized control) is satisfied follows trivially from the definition of the control law in (25). Moreover, we have already established that F1 (finite time convergence to the appropriate graph) holds trivially as long as it holds initially, and what remains to be shown here is thus that we can drive the system in finite time to a configuration in which F1 holds, after which Lemma 4.3.2 applies. Moreover, we need to establish that the inter-robot displacements (defined

for edges in E_d) converge asymptotically to the desired, relative displacements (F3), which is the topic of the next theorem.

Theorem 4.3.3. *Under the same assumptions as in corollary 4.3.2, $\|\ell_{ij}(t)\|$ converges asymptotically to $\|d_{ij}\|$ for all i, j such that $(i, j) \in E_d$.*

Proof. Based on the observation that $\mathcal{G}_d$ remains a spanning graph to the DIG, together with the observation that

$$\frac{dc(y,j)}{dt} = -\mathcal{L}_W(\Delta - \|d\|, y)c(y,j), \quad j = 1, 2, \ldots n,$$

Theorem 3.3.2 ensures that $c(y, j)$ will converge to span$\{\mathbf{1}\}$, $\forall j \in \{1, \ldots, n\}$. What this implies is that all displacements must be the same, i.e. that $y_i = \zeta$, $\forall i \in \{1, \ldots, N\}$ for some constant $\zeta \in \mathbb{R}^n$. But, this simply means that the system converges asymptotically to a fixed translation away from the target points τ_i, $i = 1, \ldots, N$, i.e.,

$$\lim_{t\to\infty} y_i(t) = \lim_{t\to\infty} \big(x_i(t) - \tau_i\big) = \zeta, \ i = 1, \ldots, N,$$

which in turn implies that

$$\lim_{t\to\infty} \ell_{ij}(t) = \lim_{t\to\infty} \big((x_i(t) - x_j(t)\big) = \lim_{t\to\infty} \big(y_i(t) + \tau_i - y_j(t) - \tau_j\big) = \zeta + \tau_i - \zeta - \tau_j = d_{ij},$$

$\forall i, j$ s.t. $(i, j) \in E_d$, which completes the proof. ∎

4.4 Hybrid, Rendezvous-to-Formation Control Strategies

The last property that must be established is that it is possible to satisfy F1, i.e. that the initial Δ-disk proximity DIG does in fact converge to a graph that has $\mathcal{G}_d$ as a spanning graph in finite time. If this was achieved then Theorem 4.3.3 would be applicable and F2 (asymptotic convergence to the correct inter-agent displacements) would follow. To achieve this, we propose to use the rendezvous control law developed in the previous section for gathering all agents into a complete graph, of which trivially

any desired graph is a subgraph. Moreover, we need to achieve this in such a manner that the assumptions in Theorem 4.3.3 are satisfied.

Let K_N denote the complete graph over N agents. Moreover, we will use K_N^ε to denote the situation in which the ε-disk proximity graph is in fact a complete graph, i.e., a DIG that is a complete graph in which no inter-agent distances are greater than ε. This notation is slightly incorrect in that graphs are inherently combinatorial objects, while inter-agent distances are geometric, and, to be more precise, we will use the notation $\mathcal{G} = K_N^\varepsilon$ to denote the fact that

$$\begin{cases} \mathcal{G} = K_N \\ \|\ell_{ij}\| \leq \varepsilon, \ \forall \ (i,j), \ i \neq j. \end{cases}$$

The reason for this construction is that, in order for Theorem 4.3.3 to be applicable, the initial condition has to satisfy $y_0 = (x_0 - \tau_0) \in \mathcal{D}^\epsilon_{\mathcal{G}_d, \Delta - \|d\|}$, which is ensured by making ε small enough. Moreover, since the rendezvous controller in (24) asymptotically achieves rendezvous, it will consequently drive the system to K_N^ε in finite time, for all ε bounded below by 0 and above by Δ.

After K_N^ε is achieved, the controller switches to the controller in (25), as depicted in Figure 9. However, this hybrid control strategy is only viable if the condition that $\mathcal{G} = K_N^\varepsilon$ is locally verifiable in the sense that the agents can decide for themselves that a synchronous mode switch is triggered [61]. In fact, if an agent has $N-1$ neighbors, i.e., degree $N-1$, all of which are within a distance $\varepsilon/2$, this implies that the maximal separation between two of those neighbors is ε. (This occurs when the agents are polar opposites on an n-sphere of radius $\varepsilon/2$.) Hence, when one agent detects this condition, it will trigger a switching signal (involving a one bit broadcast communication to all its neighbors), and the transition in Figure 9 occurs. Note, first of all, that this argument hinges on the fact that the total number of agents, N, is known to each agent. This could, arguably, be a concern for graphs with time-varying node sets. And, secondly, transition in Figure 9 might actually not occur at the exact

moment when $\mathcal{G}$ becomes K_N^ε, but rather at a later point. Regardless of which, we know that this transition will in fact occur in finite time in such a way that the initial condition assumptions of Theorem 4.3.3 are satisfied.

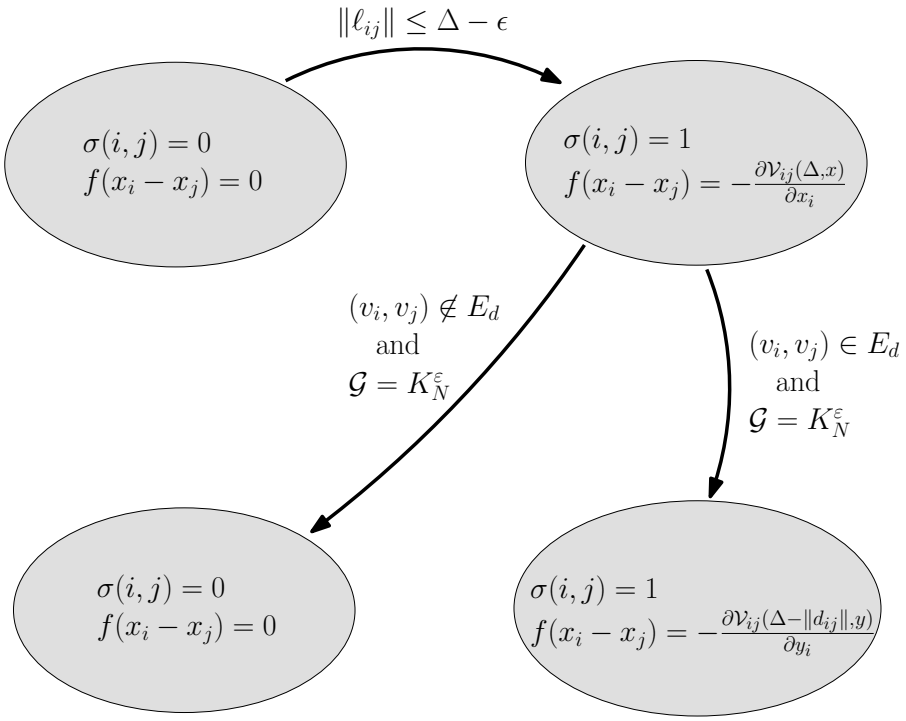


Figure 9. The figure shows a state machine describing how the system undergoes transitions from rendezvous (collection of the agents to a tight, complete graph), to formation control.

4.5 Examples

The simulation shown below highlights the proposed formation control strategy, and is implemented based on the formation control law in (25). In the simulation, five agents, starting from a straight line, are to form a pentagonal formation, with $\mathcal{G}_d = C^5$ (the cyclic graph with 5 nodes), and the desired interagent distances being $\|d_{ij}\| = 3.2$ for all $(i,j) \in E_d$. The movement of the group during the first 0.5 seconds is shown in Figure 10. The individual trajectories corresponding to the same maneuver, during the same time period, are shown in Figure 11.

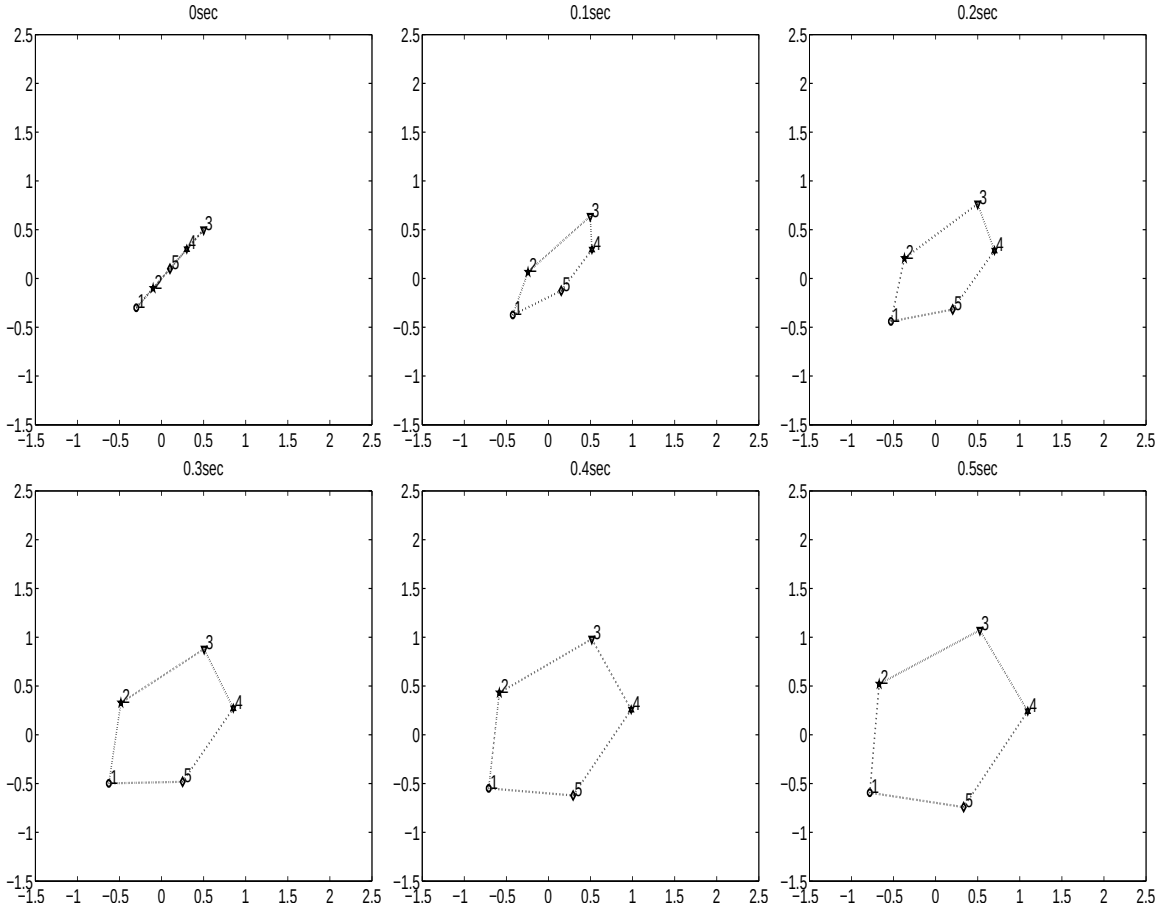


Figure 10. Evolution of the formation process.

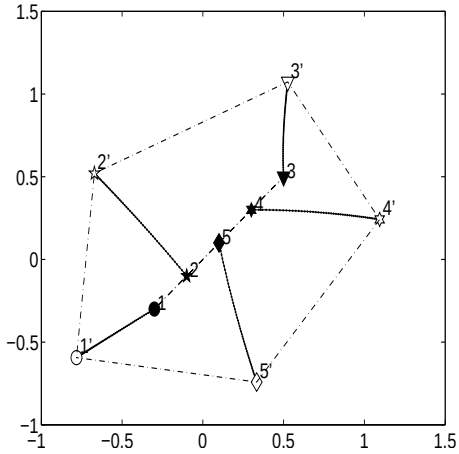


Figure 11. Trajectory of the Formation Process from t=0 sec to t=0.5 sec, starting from 1,2,3,4,5 and ending at 1',2',3',4',5'.

4.6 Conclusions

In this chapter, a of graph-based nonlinear feedback control laws is proposed for formation control of networked systems. The nonlinear feedback laws are based on a graph model of formations and the weighted graph Laplacian matrices associated to the formation graph. Furthermore, the proposed control laws are also proved to be able to guarantee that the desired connectedness is preserved during formation maneuvers. A hybrid strategy combining rendezvous and formation control is shown to be able to realize formations from arbitrary positions.

CHAPTER 5

THE FORMATION PROCESS

5.1 Introduction

Rendezvous and the formation control can be viewed as two stages of a formation process, as shown in Figure 12, where a formation is realized in three steps: first, all the agents are driven together (rendezvous); then, a position in the desired formation is assigned to each agent; and, finally, all the agents are dispatched to the target location by a formation control law.

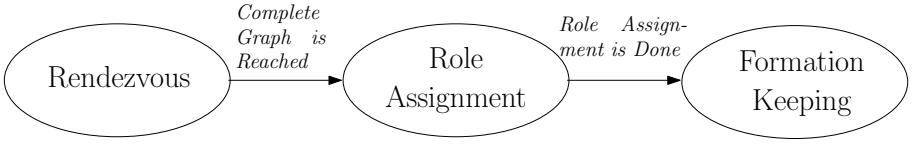


Figure 12. Three phase of a formation process: rendezvous, role-assignment, and formation

5.2 Assignment Problems

After rendezvous is achieved, it is desired to properly adjust the target formation and assign a role in it for each agent in such a way that the total distance traveled by all the agents is minimum. Since the formation defined in Chapter 4 is translationally and rotationally invariant, the problem needs to determine the *rotation, translation, and role assignment* (What roles in the target formation should the individual agents be assigned to?).

In other words, the problem considered here is to simultaneously optimize (i) the translation and rotation of the target formation and (ii) the assignment (or matching) of agents to targets. This is mathematically formulated as follows:

Let $x_1, x_2, \ldots, x_N \in \mathbb{R}^2$ be the planar positions of the agents and $y_1, y_2, \ldots, y_N \in \mathbb{R}^2$ be that of the targets. x and y denote the cascaded vector of x_i and y_i, $i = 1, \ldots, N$. We denote by $\theta \in [0, 2\pi)$ and $v \in \mathbb{R}^2$ the angular rotation and the translation of the

target formation. The assignment of the agents to the targets is described by p, which is an element of the set P_N, i.e., the set of all possible permutations over N elements. For example, $p = \{2, 3, 1\}$, with $N = 3$, means that the agents at x_1, x_2, x_3 are assigned to the targets at y_2, y_3, y_1, respectively. Furthermore, the ith element of p is represented by $p(i)$. Note that the algorithm we are seeking is not necessarily decentralized in the sense that the position information of all agents is required. This can be viewed as a "locker room agreement" after rendezvous is realized.

Now, consider the following problem:

$$\Sigma_{l_2^2}(x, y) : \min_{(v,\theta,p)\in\mathbb{R}^2\times[0,2\pi)\times P_N} J_{l_2^2}(x, y, v, \theta, p), \tag{30}$$

$$\text{s.t. } J_{l_2^2}(x, y, v, \theta, p) = \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)} + v)\|_2^2. \tag{31}$$

Theorem 5.2.1. *Suppose that $N \in \mathbb{N}$, $x_1, x_2, \ldots, x_N \in \mathbb{R}^2$, and $y_1, y_2, \ldots, y_N \in \mathbb{R}^2$ are given. Let (v^*, θ^*, p^*) denote a globally optimal solution to $\Sigma_{l_2^2}(x, y)$. Then, the following holds.*

(i) The optimal translation is

$$v^* = R(\theta^*)^T x_c - y_c, \tag{32}$$

where $x_c \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i$ and $y_c \triangleq \frac{1}{N}\sum_{i=1}^{N} y_i$ are the centers of mass of the agent and target positions, respectively.

(ii) The optimal solution to the problem

$$\min_{\theta\in[0,2\pi)} J_{l_2^2}(x, y, v^*, \theta, p^*) \tag{33}$$

is

$$\theta^* = \tan^{-1}\left(\frac{W_2(v^*, p^*)}{W_1(v^*, p^*)}\right), \tag{34}$$

where

$$W_1(v,p) \triangleq \sum_{i=1}^{N} x_i^T (y_{p(i)} + v),$$

$$W_2(v,p) \triangleq \sum_{i=1}^{N} x_i^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (y_{p(i)} + v).$$

(iii) The optimal (possibly not unique) assignment satisfies

$$p^* = \arg\min_{p \in P_N} J_{l_2^2}(x, y, v^*, \theta^*, p). \tag{35}$$

Moreover, the problem

$$\min_{p \in P_N} J_{l_2^2}(x, y, v^*, \theta^*, p) \tag{36}$$

corresponds to the linear assignment problem.

Proof. (i) Since

$$\begin{aligned} J_{l_2^2}(x,y,v,\theta,p) &= \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)} + v)\|_2^2 \\ &= \sum_{i=1}^{N} \big[(R(\theta)^T x_i - y_{p(i)} - v)^T R(\theta)^T \\ &\quad R(\theta)(R(\theta)^T x_i - y_{p(i)} - v)\big] \\ &= \sum_{i=1}^{N} \big[\big(v - (R(\theta)^T x_i - y_{p(i)})\big)^T \\ &\quad \big(v - (R(\theta)^T x_i - y_{p(i)})\big)\big], \end{aligned}$$

the derivative of $J_{l_2^2}(x,y,v,\theta,p)$ with respect to v is given by

$$\begin{aligned} \frac{\partial J_{l_2^2}}{\partial v} &= \sum_{i=1}^{N} 2\big(v - (R(\theta)^T x_i - y_{p(i)})\big)^T \\ &= 2N\big(v - (R(\theta)^T x_c - y_c)\big)^T. \end{aligned}$$

It should be noted that this derivative does not depend on the assignment p. Hence, by noting that $v^* = \arg\min_{v \in \mathbb{R}^2} J_{l_2^2}(x,y,v,\theta^*,p)$ for any p, and that $J_{l_2^2}$ is convex in

v, we obtain

$$v^* - (R(\theta^*)^T x_c - y_c) = 0 \tag{37}$$

as the first order necessary condition for the problem $\min_{v\in\mathbb{R}^2} J_{l_2^2}(x, y, v, \theta^*, p^*)$, which proves (32).

(ii) From

$$\|x_i - R(\theta)(y_{p(i)} + v)\|_2^2 = x_i^T x_i - 2x_i^T R(\theta)(y_{p(i)} + v) + (y_{p(i)} + v)^T (y_{p(i)} + v),$$

it follows that

$$\theta^* = \underset{\theta\in[0,2\pi)}{\arg\min} \sum_{i=1}^{N} -x_i^T R(\theta)(y_{p^*(i)} + v^*). \tag{38}$$

In addition, we have

$$\begin{aligned}
&\sum_{i=1}^{N} -x_i^T R(\theta)(y_{p(i)} + v) \\
&= -\sum_{i=1}^{N} x_i^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} (y_{p(i)} + v)\cos\theta + x_i^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (y_{p(i)} + v)\sin\theta \\
&= -(W_1(v, p)\cos\theta + W_2(v, p)\sin\theta),
\end{aligned}$$

and hence (34) follows.

(iii) Per definition, (35) holds. In addition, by defining $\tilde{y}_{p(i)} \triangleq R(\theta^*)(y_{p(i)} + v^*)$ we obtain

$$\min_{p\in P_N} J_{l_2^2}(x, y, v^*, \theta^*, p) = \min_{p\in P_N} \sum_{i=1}^{N} \|x_i - \tilde{y}_{p(i)}\|^2,$$

which in turn implies that (36) corresponds to the linear assignment problem. (See [63] for further details of the linear assignment problem.) ∎

What Theorem 5.2.1 means is that we can solve the problem $\Sigma_{l_2^2}$ if two of the three optimal parameters v^*, θ^*, or p^* are provided. In fact, (i) implies that v^* does not depend upon p^* and can be obtained by (32) if θ^* is provided. What (ii)

shows is that θ^* is given by (34) if v^* and p^* are given. Moreover, (iii) implies that since (36) is a linear assignment problem, p^* is easily computed, e.g., using the Hungarian method, which is a polynomial time algorithm whose computational complexity is $O(N^3)$ [63]. However, the problem of solving for these three parameters simultaneously is not trivial. As the problem becomes more complex when we want to optimize over the three parameters simultaneously, we seek some feasible way that can lead us to suboptimal, yet reasonably good solutions to the problem (30).

Note that in Theorem 5.2.1, (36) and (33) correspond to (30) with fixed θ and with fixed p, respectively, which implies that if either θ or p is fixed, (30) can be solved. More precisely, (33) is explicitly solved and (36) can be efficiently solved using well-known methods for the linear assignment problem, e.g., using the Hungarian method.

First, observe that since v^* is independent of p, and in fact given by

$$v^*(\theta) = R^T(\theta)x_c - y_c,$$

we can express the cost $J_{l_2^2}(x, y, v, \theta, p)$ without reference to v through

$$J_{l_2^2}(x, y, \theta, p) = \sum_{i=1}^{N} \|x_i - R(\theta)(y_{p(i)} + R^T(\theta)x_c - y_c)\|_2^2.$$

Now, note that

$$\begin{aligned} & x_i - R(\theta)(y_{p(i)} + R^T(\theta)x_c - y_c) \\ = \; & x_i - R(\theta)y_{p(i)} - x_c + R(\theta)y_c) \\ = \; & x_i - x_c - R(\theta)(y_{p(i)} - y_c). \end{aligned}$$

Hence, with a slight abuse of notation, we can assume that x_c and y_c have already been absorbed by the state variables. In other words, we let $x_i \triangleq x_i - x_c$ and $y_{p(i)} \triangleq y_{p(i)} - y_c$, which corresponds to the original and target formations whose center of mass is equal to the origin. Since the decision variables here are p and θ, for simplicity reason, we will denote the cost function as $J(p, \theta)$ for a given x and y, if it is clear from context.

5.3 Numerical Method for a Suboptimal Solution

Based on Theorem 5.2.1, together with the above observation, we propose four methods. In what follows, $(\theta_i^{\#}, p_i^{\#})$ $(i = 1, 2, 3, 4)$ represent the corresponding suboptimal solutions.

Method A: Arbitrary Initial Rotation

In this method, we start from a target formation with 0 initial rotation, i.e., $\theta_{initial} = 0$, and find the resulting optimal assignment of $p_1^{\#}$. Then, based on that assignment, we find the optimal rotation angle $\theta_1^{\#}$, i.e.,

$$\begin{cases} p_1^{\#} & \triangleq & p^*(0), \\ \theta_1^{\#} & \triangleq & \theta^*(p_1^{\#}). \end{cases} \tag{39}$$

Rather than producing a particularly good solution, this simple method gives us a basic building block from which we can construct more sophisticated methods, leading to better results. One way to compose a better method is to repeat *Method A*, which leads to the next method.

Method B: Iterative Method

Another possible approach for obtaining a practical solution is to mutually and iteratively apply Theorem 5.2.1 (i) and (ii). In other words, repeat *Method A* until its solution converges. The solution is thus given by

$$\begin{cases} p_2^{\#} & \triangleq & p_2^{\#}(N_{iter}) \\ \theta_2^{\#} & \triangleq & \theta_2^{\#}(N_{iter}), \end{cases} \tag{40}$$

where N_{iter} is the total number of the iterations, $\theta_2^{\#}(i)$ and $p_2^{\#}(i)$, for $i = 1, 2, \ldots, N_{iter}$, are defined as $p_2^{\#}(0) \triangleq p^*(0)$, $\theta_2^{\#}(0) \triangleq \theta^*(p_2^{\#}(0))$, and

$$\begin{cases} p_2^{\#}(i) & \triangleq & p^*(\theta_2^{\#}(i-1)) \\ \theta_2^{\#}(i) & \triangleq & \theta^*(p_2^{\#}(i)). \end{cases} \tag{41}$$

Using this method, we expect substantial improvements in the solution. At the same time, the computational cost increases linearly in N_{iter}. However, because of

the discrete and finite set P over which p takes values, it is unclear if this method converges, and if so, to what accumulation point. Note that the notion of a local minimum is ill-defined since P is not a topological space.

Method C: Angular Discretization

The solution $(\theta_3^{\#}, p_3^{\#})$ is given based on the discretization of the rotation angles,

$$\begin{cases} p_3^{\#} & \triangleq & p^*(\theta_3^{\#}) \\ \theta_3^{\#} & \triangleq & \arg\min_{\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}} J(x, y, \theta, p^*(\theta)), \end{cases} \tag{42}$$

i.e., $(\theta_3^{\#}, p_3^{\#})$ is an optimal solution to the problem

$$\min_{\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}} \min_{p \in P_N} \sum_{i=1}^{N} \|x_i - R(\theta) y_{p(i)}\|^2. \tag{43}$$

Note that we calculate an optimal solution to (43) by solving (36) for every $\theta \in \{\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_{d-1}\}$, where $\theta^*(p)$ and $p^*(\theta)$ are defined in Theorem 5.2.1 and $\hat{\theta}_i \triangleq 2\pi k/d$ $(k = 0, 1, \ldots, d-1)$ for a given positive integer d.

Method D: Improved Angular Discretization

This is an extension of Method C in the sense that after finding the optimal assignment $p^*(\theta_3^{\#})$, we proceed with one more step of optimization in which we find the optimal angle with respect to $p^*(\theta_3^{\#})$. Hence,

$$\begin{cases} p_4^{\#} & \triangleq & p^*(\theta_3^{\#}) \\ \theta_4^{\#} & \triangleq & \theta^*(p_4^{\#}). \end{cases} \tag{44}$$

As four methods are proposed, now, we need to examine the following:

- Are the results close enough to the optimal solution?
- How much time dose each method take?

Table 1 and Figure 5.3 show the simulation results for $N = 8$, where we take $d \triangleq 100$ for obtaining $(\theta_3^{\#}, p_3^{\#})$ and $(\theta_4^{\#}, p_4^{\#})$. Moreover, the problem (36) is solved

by the Hungarian method, where the optimal solution obtained by enumerating all elements of P_N is also shown (note $\mathrm{card}(P_N) = 40320$).

Table 1. Result of numerical simulations.

i	1	2($N_{iter} \triangleq 30$)	3	4	optimal
$\theta_i^{\#}$ [rad]	-0.216	-1.3339	-1.916	-1.931	-1.931
$\{\mathrm{i} \to p_i^{\#}\}$	$\begin{Bmatrix} 1\to 8 \\ 2\to 3 \\ 3\to 7 \\ 4\to 1 \\ 5\to 4 \\ 6\to 2 \\ 7\to 5 \\ 8\to 6 \end{Bmatrix}$	$\begin{Bmatrix} 1\to 8 \\ 2\to 3 \\ 3\to 6 \\ 4\to 4 \\ 5\to 7 \\ 6\to 5 \\ 7\to 2 \\ 8\to 1 \end{Bmatrix}$	$\begin{Bmatrix} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{Bmatrix}$	$\begin{Bmatrix} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{Bmatrix}$	$\begin{Bmatrix} 1\to 6 \\ 2\to 8 \\ 3\to 5 \\ 4\to 4 \\ 5\to 7 \\ 6\to 1 \\ 7\to 3 \\ 8\to 2 \end{Bmatrix}$
$J(\theta_i^{\#}, p_i^{\#})$	63.6973	33.9484	32.3675	32.3529	32.3529
Computation time [sec]	0.016	0.3900	0.922	0.938	260.203

We can see that $(\theta_3^{\#}, p_3^{\#})$ and $(\theta_4^{\#}, p_4^{\#})$ are better than the other solutions in terms of their costs. On the other hand, $(\theta_1^{\#}, p_1^{\#})$ is better from the viewpoint of the computation time, while $(\theta_2^{\#}, p_2^{\#})$ might be the best compromise between accuracy and computation time. As the number of iterations is set to 30, the evolution of the solution is shown in Fig. 14. It is worth noticing that the solution is not improving after some iterations since *method B* does not guarantee a global optimal solution.

Finally, we apply *method D* to the originally posed three-phase problem of sequential *rendezvous*, *assignment*, and *dispatch*. An example of this is shown in Fig. 15, 16 and 17.

Remark 5.3.1. *method D* lead to the global solution if the discretization of θ is fine enough. In other words, as d becomes substantially large, the global minimum will be obtained. This observation could be turned into an algorithm (*method D'*), where d is iteratively increased. However, it is premature to declare that *method D'* is a numerically tractable algorithm, because the problem of choosing a lower bound on d is not known.

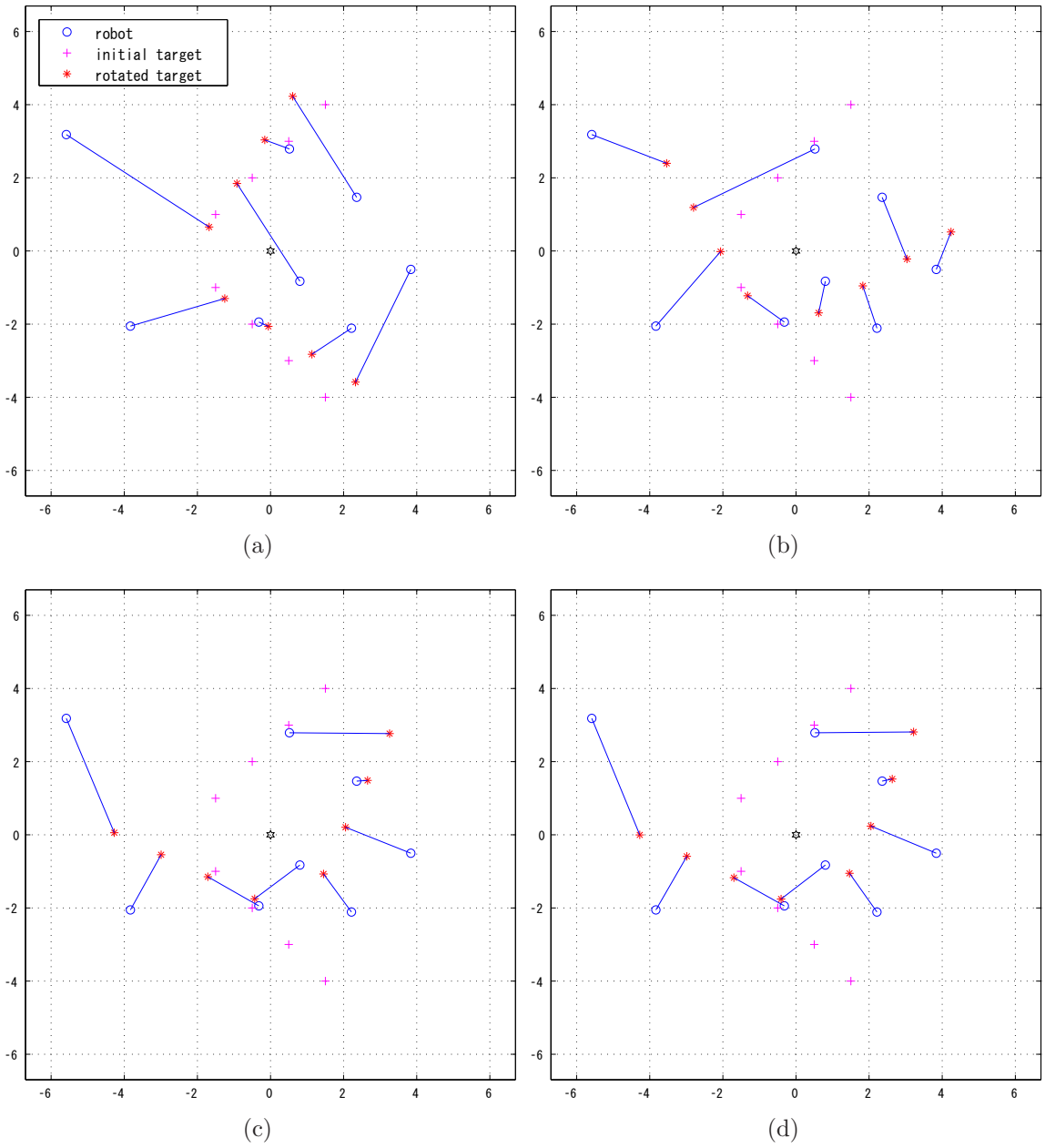


Figure 13. Result of numerical simulations: (a) Method A: $(\theta_1^{\#}, p_1^{\#})$; (b) Method B: $(\theta_2^{\#}, p_2^{\#})$; (c) Method C: $(\theta_3^{\#}, p_3^{\#})$; (d) Method D: $(\theta_4^{\#}, p_4^{\#})$.

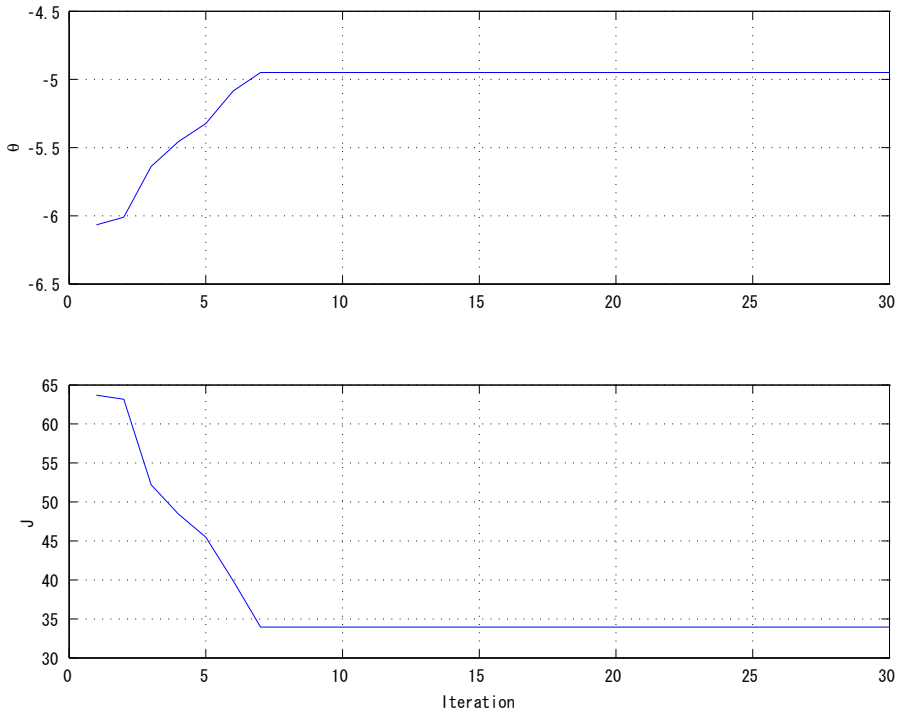


Figure 14. Changes of $\theta_2^{\#}(i)$ and $J(\theta_2^{\#}(i), p_2^{\#})$ in method B. Depicted is the rotation angle as a function of the iteration number (upper figure) together with the corresponding cost (lower figure).

5.4 Discussions and Conclusions

A formation process contains three steps: rendezvous, role assignment, and formation. As a connection of the results from previous chapters, this chapter studies the problem of which role an agent should take in a translationally and rotationally invariant formation. It has been shown that with any two variables fixed, the third variable can be solved analytically. Following this observation, we have proposed four numerical methods which can achieve suboptimal solutions, and compared their results.

Now, let us discuss some issues pertaining to the computational complexity associated with the different assignment algorithms. As shown in [63], the problem (36) is solved by the Hungarian method using $O(N^3)$ operations, which is higher than the complexity associated with (33). This is the case since in order to solve (33) the number of necessary function evaluations is linear in N.

We now let N_a^i $(i = 1, 2, 3, 4)$ denote the number of assignments that must be

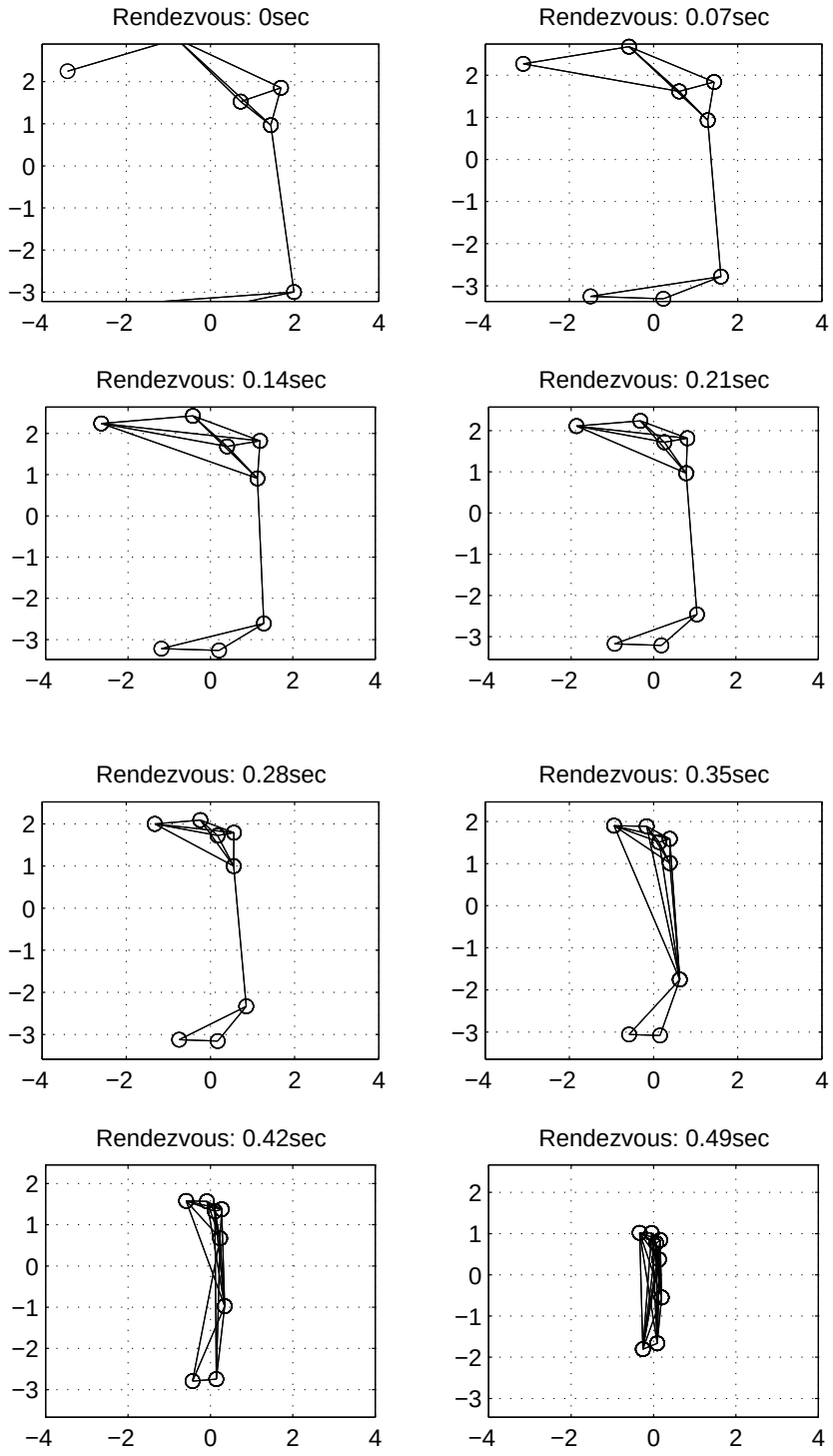


Figure 15. Phase I: The rendezvous procedure, starting from an arbitrary, connected graph, generate a complete graph after 0.45 second.

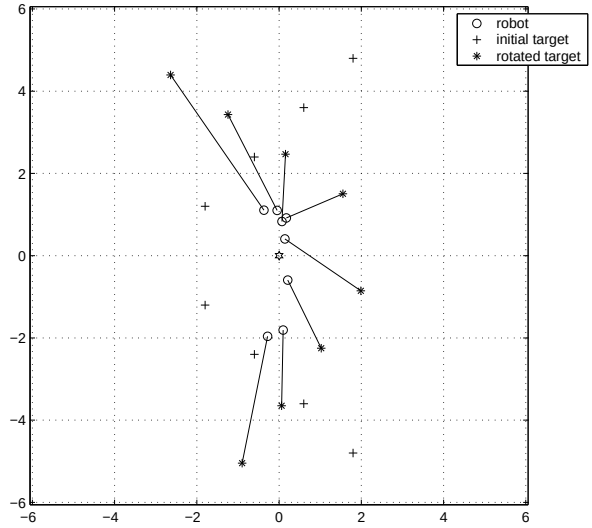


Figure 16. Phase II: A suboptimal assignment is obtained using method D.

solved in order to compute the four suboptimal solutions in (39)-(44). It is straightforward to show that

$$N_a^1 = 1,\ N_a^2 = N_{iter},\ N_a^3 = d,\ N_a^4 = d + 1. \tag{45}$$

Hence, the computational complexity associated with the best of the four methods is $O(N^3 d)$, which is certainly less than the $O(N!)$ obtained through permutation enumeration.

The question now is how to choose d in such a way that the solution to the problem in (44) approaches the solution to the original problem (30) as closely as possible. To this end we let n_N be the average number of distinctly different assignments encountered as θ sweeps through d values, as $d \gg 1$. The average is obtained by generating a large number of random formations of N agents. In Figure 18 we have plotted n_N as a function of N and it appears that n_N is linear in N. What this means is that d should be linear in N in order to obtain an adequate solution, which implies that the complexity becomes $O(N^4)$. However this is not sufficient to determine the complexity of the problem since one need to ensure that d is large enough to capture the correct assignments. And to find this d is certainly not an easy task.

Finally, it should be noted that since the computations must be computed across

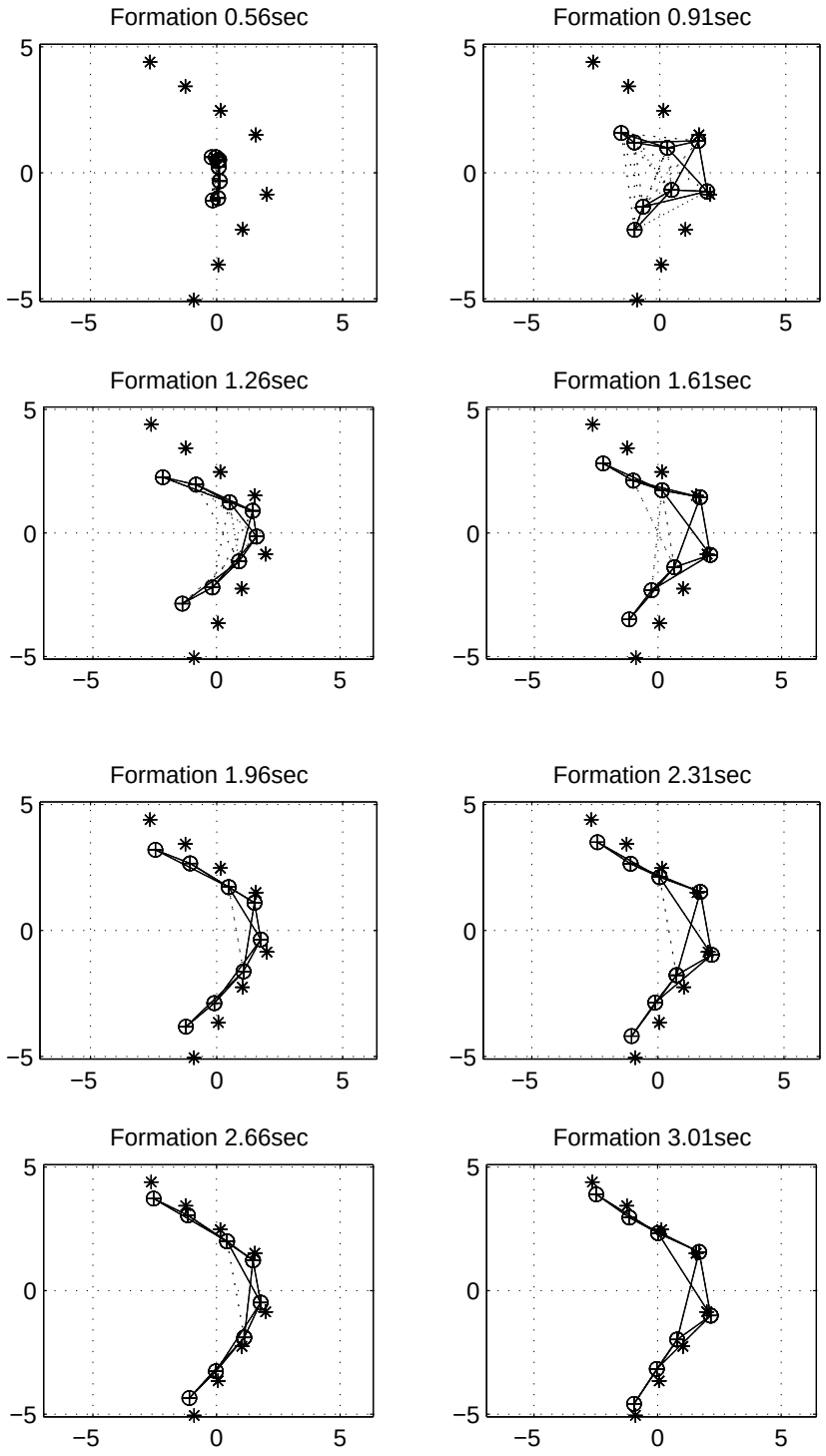


Figure 17. Phase III: Target formation is achieved after 2.4 seconds. In the graph, circles denote the actual positions of the agents, while asterisks denote the target positions.

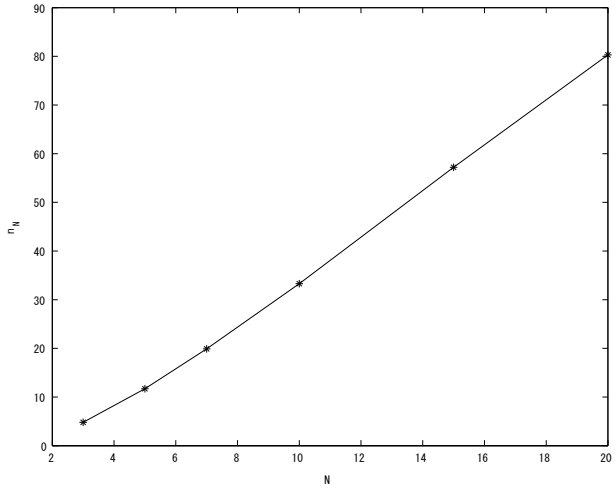


Figure 18. Number of assignment related to N

the different agents, further complexity reductions should be possible through decentralization and/or parallelization of these computations. This work is, however, left for future investigation.

CHAPTER 6

DISTRIBUTED CONTROL OF HETEROGENEOUS NETWORKED SYSTEMS

6.1 Introduction

The situations to be considered in this chapter are where some nodes in the network do not abide by the agreement protocol. We refer to these agents as *leaders*. The nods in the network other than the leaders are referred to as followers. We call this network structure, where nodes are divided into a leader set and a follower set, leader-follower structures. Different behavior presented in such a structure, when the leaders take control, is the major topic of this chapter. Particularly, we want to provide a graph-theoretic characterization of the controllability of certain leader-based, multi-agent systems.

The controllability issue in leader-follower multi-agent systems was first introduced by Tanner in [42], where a necessary and sufficient condition for controllability was given based on the eigenvectors of the graph Laplacian. Although elegant, this condition was not graph-theoretic because controllability could not be directly decided from the graph topology itself. A more topological result was given by Mesbahi and Rahmani [43], in which a sufficient condition for the network to be uncontrollable in the case of one anchored (leader) agent was given. Their result was related to the symmetry and automorphism group of the underlying graph. In [44], we proposed a necessary condition in terms of cut space of a graph. In this thesis, we further extend this notion and present a more general condition based on so-called equitable partitions of the underlying graph. Our result thus addresses a scenario where multiple leaders are possible, and from an equitable partition point-of-view, captures a larger set of graph topologies.

6.2 Leader-Follower Structure

In a networked system, we can easily imagine a subset of the agents as having superior sensing and communication abilities. These advanced agents are required to take leader roles, while the others follow them. In this chapter, we investigate the control method of this structure and present a graph theoretical interpretation of an important issue, controllability of a leader-follower network. In what follows, we use subscript l to denote the affiliation with leaders while f for the followers. For example, a *follower graph* $\mathcal{G}_f$ is the subgraph induced by the follower set V_f. As the leader roles are designated, the incidence matrix $\mathcal{B}$ can be partitioned as

$$\mathcal{B} = \begin{bmatrix} \mathcal{B}_f \\ \mathcal{B}_l \end{bmatrix},$$

where $\mathcal{B}_f \in \mathbb{R}^{N_f \times M}$, and $\mathcal{B}_l \in \mathbb{R}^{N_l \times M}$. Here N_f, N_l and M are the cardinalities of the follower group, the leader group, and the edge set respectively. As a result, the graph Laplacian $\mathcal{L}$ is given by

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_f & l_{fl} \\ l_{fl}^T & \mathcal{L}_l \end{bmatrix}, \tag{46}$$

where

$$\mathcal{L}_f = \mathcal{B}_f \mathcal{B}_f^T,\ \mathcal{L}_l = \mathcal{B}_l \mathcal{B}_l^T \text{ and } l_{fl} = \mathcal{B}_f \mathcal{B}_l^T.$$

As an example, Figure 19 shows a leader-follower network with $V_l = \{5, 6\}$ and $V_f = \{1, 2, 3, 4\}$. This gives

$$\mathcal{B}_f = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathcal{B}_l = \left[\begin{array}{c|cccc} \mathbf{0}_{2\times 4} & \begin{matrix} 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{matrix} \end{array}\right],$$

and

$$\mathcal{L}_f = \begin{bmatrix} 3 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ -1 & 0 & -1 & 3 \end{bmatrix}, \; l_{fl} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}.$$

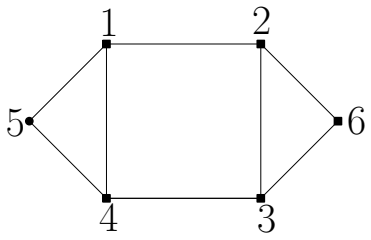


Figure 19. A leader-follower network with: $V_f = \{1, 2, 3, 4\}$ **and** $V_l = \{5, 6\}$**.**

Now, recall the Laplacian-based control law given in Chapter 3 and assume the state of each agent is given by a scalar[1], then the system given by 10 can be simplified as

$$\dot{x} = -\mathcal{L}(\mathcal{G})x; \tag{47}$$

The system we are interested in is the leader-follower system, where the followers are governed by the Laplacian-based feedback law, while the leaders' movements are dictated by some exogenous control signals. First, let the exogenous signals be zero, the leader-follower system can be written as

$$\begin{bmatrix} \dot{x}_f \\ \dot{x}_l \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_f(\mathcal{G}) & l_{fl}(\mathcal{G}) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_f \\ x_l \end{bmatrix}, \tag{48}$$

or simply

$$\dot{x}_f \;\; = \;\; -\mathcal{L}_f(\mathcal{G})x_f - l_{fl}(\mathcal{G})x_l, \tag{49}$$

Moreover, let us assume the network topology does not change, i.e., we consider a static interaction graph (SIG). Under this setup, a natural question is as follows:

[1]The argument in one-dimension case can be easily extend to decoupled multi-dimension case.

Can the leader(s) move the followers to any desired configuration? From a control theoretic point of view, we need to answer following questions:

1. Is this system controllable?
2. If not, what is the controllability decomposition?
3. What is the controllable subspace?
4. What is the structural significance of the controllable or uncontrollable subspace?

Answers to these questions are provided in the next section.

6.3 Controllability Analysis of the Leader-Follower Structure

Before we provide conditions for a multiple-leader networked system to be controllable, let us review some basic controllability results [64].

Given $x \in \mathbb{R}^N,\ u \in \mathbb{R}^M,\ A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times M}$, the system $\dot{x} = Ax + Bu$ is controllable if and only if, $\forall\ v_i \in \text{spec}(A)$, where $\text{spec}(\cdot)$ denotes the spectrum, $v_i \notin \mathcal{N}(B^T)$. Specifically, for the system $(-\mathcal{L}_f, -l_{fl})$, the following statements are equivalent:

1. The system is completely controllable;
2. None of the eigenvectors[2] of $\mathcal{L}_f$ is in the nullspace of l_{fl}^T, i.e., $\forall\ v_i$, such that $\mathcal{L}_f v_i = \lambda v_i$ for some $\lambda \in \mathbb{R}$, $v_i \notin \text{Null}(l_{fl}^T)$;
3. The controllability matrix $\mathcal{C}_{-\mathcal{L}_f, -l_{fl}}$ has full rank;
4. The matrix $[\lambda I - \mathcal{L}_f \mid l_{fl}]$ has full rank for all $\lambda \in \mathbb{R}$.

Based on these results, together with some properties of the graph Laplacian, we can derive the following lemma.

[2]Since $\mathcal{L}_f$ is symmetric, its left eigenvectors are equal to the right ones.

Lemma 6.3.1. *Given a connected graph, the system $(-\mathcal{L}_f, -l_{fl})$ is controllable if and only if $\mathcal{L}$ and $\mathcal{L}_f$ do not share any common eigenvalues.*

Proof of Necessity: [3]

We can reformulate the lemma as stating that the system is uncontrollable if and only if there exists at least one common eigenvalue between $\mathcal{L}$ and $\mathcal{L}_f$. First, we show the necessity. Suppose the system is uncontrollable. Then there exists a vector $v_i \in \mathbb{R}^{n_f}$ such that $\mathcal{L}_f v_i = \lambda v_i$ for some $\lambda \in \mathbb{R}$, with

$$l_{fl}^T v_i = \mathbf{0}.$$

Now, since

$$\begin{bmatrix} \mathcal{L}_f & l_{fl} \\ l_{fl}^T & \mathcal{L}_l \end{bmatrix} \begin{bmatrix} v_i \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_f v_i \\ l_{fl}^T v_i \end{bmatrix} = \lambda \begin{bmatrix} v_i \\ \mathbf{0} \end{bmatrix},$$

λ is also an eigenvalue of $\mathcal{L}$, with eigenvector $[v_i^T, \mathbf{0}]^T$. ∎

6.3.1 A Sufficient Condition

We first approach the controllability problem of the leader-follower structure from the null-space of the incidence matrix of the graph.

Lemma 6.3.2. *If $\mathcal{G}$ is connected, then $\mathcal{L}_f$ is positive definite.*

Proof. It is well known that $\mathcal{L}(\mathcal{G}) \succeq 0$. In addition, if $\mathcal{G}$ is connected, we have that $\text{Null}(\mathcal{L}(\mathcal{G})) = \text{span}\{\mathbf{1}\}$, where $\text{Null}(\cdot)$ denotes the null space and $\mathbf{1}$ is the vector with all entries being one. Now, since

$$x_f^T \mathcal{L}_f x_f = [x_f^T \ \mathbf{0}] \mathcal{L} \begin{bmatrix} x_f \\ \mathbf{0} \end{bmatrix}$$

[3] The sufficiency proof will be given after Lemma 6.3.9.

and $[x_f^T\ \mathbf{0}]^T \notin \mathcal{N}(\mathcal{G})$, we have that

$$[x_f^T\ \mathbf{0}]\mathcal{L}\begin{bmatrix} x_f \\ \mathbf{0} \end{bmatrix} > 0\ \forall\ x_f \in \mathbb{R}^{N_f},$$

and the lemma follows. ∎

Consider the control law given in (49), we can derive the location of the followers when the leaders are semi-static[4].

Theorem 6.3.3. *Given a fixed leader position x_l, the semi-static equilibrium position of the followers under dynamics in* (49) *is*

$$x_f = -\mathcal{L}_f^{-1} l_{fl} x_l, \tag{50}$$

which is globally asymptotically stable.

Proof. From the previous Lemma 6.3.2, we know that $\mathcal{L}_f$ is invertible and hence (50) is well-defined. Since, at the unique stable equilibrium point, we have

$$\begin{aligned} \dot{x}_f &= -\mathcal{L}_f x_f - l_{fl} x_l \\ &= -\mathcal{L}_f (x_f + \mathcal{L}_f^{-1} l_{fl} x_l) = 0. \end{aligned} \tag{51}$$

Moreover, since $\mathcal{L}_f \succ 0$ this implies global asymptotic stability. ∎

Theorem 6.3.4. *The system $(-\mathcal{L}_f, -l_{fl})$ is controllable if $\mathcal{G}$ is connected and $\mathcal{N}(\mathcal{B}_l) \subseteq \mathcal{N}(\mathcal{B}_f)$.*

Proof. For the system $(-\mathcal{L}_f, -l_{fl})$, being controllable means that $v_i(\mathcal{L}_f) \notin \mathcal{N}(l_{fl}^T)$, $\forall\ v_i \in \text{spec}(\mathcal{L}_f)$. In other words, $\mathcal{B}_l \mathcal{B}_f^T v_i \neq 0$. Thus if $\mathcal{N}(\mathcal{B}_l) \subseteq Im(\mathcal{B}_f^T)^\perp = \mathcal{N}(\mathcal{B}_f)$, the system is controllable. ∎

[4]This means the leaders are moving in a very slow fashion such that, at each time instance, the system is at equilibrium state.

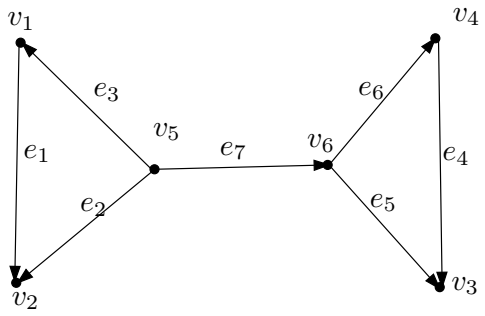


Figure 20. An example of choosing leaders such that condition in Theorem 6.3.4 is satisfied.

Note that as a consequence of Theorem 6.3.4, we have a constructive way of assigning leadership roles to agents to ensure controllability.

Given a network topology, first find the null space of $\mathcal{B}$. Then select the appropriate rows of $\mathcal{B}$ and stack them in a new matrix such that the null space of this new matrix is embedded in $\mathcal{N}(\mathcal{B})$. As an example, consider the directed graph in Figure 20, here we have

$$\mathcal{B} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$$

with

$$\mathcal{N}(\mathcal{B}) = \text{span} \left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} \right\}$$

From the incidence matrix we directly see that by choosing any single agent as a

follower and the remaining five as leaders, Theorem 6.3.4 will be satisfied.

It is worth noticing that this sufficient condition is conservative. In many cases we can find configurations with fewer leaders that are still controllable. For instance, in Figure 20, we can choose nodes v_1, v_2, v_3 and v_4 as leaders and the system is still controllable. We can derive a more strict necessary condition with the help of some new graph theoretical instruments.

6.3.2 Interlacing and Equitable Partitions of Graphs

Equitable partitions and interlacing theory play an important role in deriving the necessary condition in the following section. In this section, we introduce some definitions and lemmas needed to support the next section.

Definition 6.3.1. A r-partition π of $V(\mathcal{G})$, with cells $C_1, \ldots, C_r$, is said to be *equitable* if each node in C_j has the same number of neighbors in C_i, for all i, j. We denote the cardinality of the partition π with $r = |\pi|$.

Let b_{ij} be the number of neighbors in C_j of a node in C_i. The directed graph with the r cells of π as its nodes and b_{ij} edges from the ith to the jth cells of π is called the *quotient* of $\mathcal{G}$ over π, and is denoted by $\mathcal{G}/\pi$. An obvious trivial partition is the n-partition, $\pi = \{\{1\}, \{2\}, \ldots, \{n\}\}$. If a partition contains at least one cell with more than one node, we call it a *nontrivial equitable partition* (NEP), and the *adjacency matrix of the quotient* is given by

$$\mathcal{A}(\mathcal{G}/\pi)_{ij} = b_{ij}.$$

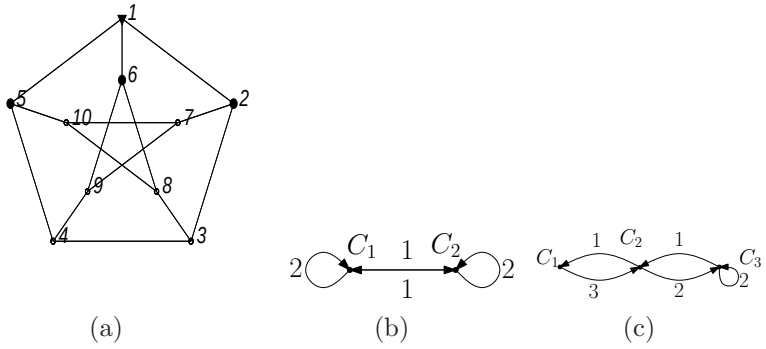
The equitable partition can be derived from graph automorphisms. For example, in the so-called Peterson graph, shown in Figure 21(a), one equitable partition π_1 (Figure 21(b)) is given by the two orbit of the automorphism groups, namely the 5 inner vertices and the 5 outer vertices. The adjacency matrix of the quotient is given

by

$$\mathcal{A}(\mathcal{G}/\pi_1) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

The equitable partition can also be introduced by the equal distance partition. Let $C_1 \subset V(\mathcal{G})$ be a given cell, and let $C_i \subset V(\mathcal{G})$ be the set of vertices at distance $i-1$ from C_1. C_1 is said to be *completely regular* if its distance partition is equitable. For instance, every vertex in the Peterson graph is completely regular and introduces the partition π_2 as shown in Figure 21(c). The adjacency matrix of this quotient is given by

$$\mathcal{A}(\mathcal{G}/\pi_2) = \begin{bmatrix} 0 & 3 & 0 \\ 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix}.$$



(a) (b) (c)

Figure 21. Example of equitable partitions on (a) the Peterson graph $\mathcal{G} = J(5,2,0)$ and the quotients: (b) the NEP introduced by the automorphism is $\pi_1 = \{C_1^1, C_2^1\}, C_1^1 = \{1,2,3,4,5\}, C_2^1 = \{6,7,8,9,10\}$, and (c) the NEP introduced by equal-distance partition is $\pi_2 = \{C_1^2, C_2^2, C_3^2\}, C_1^2 = \{1\}, C_2^2 = \{2,5,6\}, C_3^2 = \{3,4,7,8,9,10\}$.

The adjacency matrix of the original graph and the quotient are closely related through the so-called interlacing theorem. First, we introduce the following lemma.

Definition 6.3.2. A *characteristic vector* $p_i \in \mathbb{R}^n$ of a nontrivial cell C_i has 1's in the positions associated with C_i and 0's elsewhere. A *characteristic matrix* $P \in \mathbb{R}^{n\times r}$of a partition π of $V(\mathcal{G})$ is a matrix with the characteristic vectors of the cells as its columns. For example, the characteristic matrix of the equitable partition of the

graph in Figure 22(a) is given by

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{52}$$

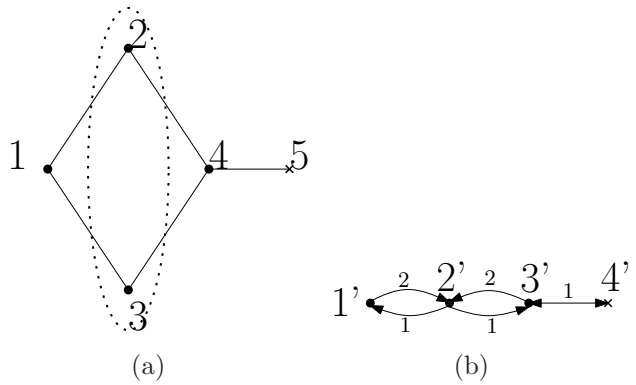


Figure 22. The (a) equitable partition and (b) the quotient of a graph.

Lemma 6.3.5. *([57] Lemma 9.3.1) Let P be the characteristic matrix of an equitable partition π of the graph $\mathcal{G}$, and let $\hat{\mathcal{A}} = \mathcal{A}(\mathcal{G}/\pi)$. Then $\mathcal{A}P = P\hat{\mathcal{A}}$ and $\hat{\mathcal{A}} = P^{+}\mathcal{A}P$, where $P^{+} = (P^{T}P)^{-1}P^{T}$ is the pseudo-inverse of P.*

As an example, the graph in Figure 22 has a nontrivial cell $(2, 3)$. The adjacency matrix of original graph is

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The adjacency matrix of the quotient is

$$\hat{\mathcal{A}} = P^{+}\mathcal{A}P = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Lemma 6.3.6. *([57] Lemma 9.3.2) Let $\mathcal{G}$ be a graph with adjacency matrix $\mathcal{A}$, and let π be a partition of $V(\mathcal{G})$ with characteristic matrix P, then π is equitable if and only if the column space of P is $\mathcal{A}$-invariant.*

Lemma 6.3.7. *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and let S be a subspace of $\mathbb{R}^n$. Then $S^{\perp}$ is A-invariant if and only if S is A-invariant.*

The proof of this well-known fact can for example be found in [65].

Remark 6.3.1. Let $\mathcal{R}(\cdot)$ denote the range space. Suppose $|V(\mathcal{G})| = n$, $|C_i| = n_i$ and $|\pi| = r$, then we can find an orthogonal decomposition for $\mathbb{R}^n$ as

$$\mathbb{R}^n = \mathcal{R}(P) \oplus \mathcal{R}(Q), \tag{53}$$

where the matrix Q satisfies $\mathcal{R}(Q) = \mathcal{R}(P)^{\perp}$, such that its columns together with those of P form a basis for $\mathbb{R}^n$. Following Lemma 6.3.7, $\mathcal{R}(Q)$ is also $\mathcal{A}-$invariant.

Unlike matrix P, Q is derived from the nullspace of P and can be constructed in different ways. One possible choice of such a Q is the $n \times n-r$ matrix with r column blocks $Q = [Q_1, Q_2, \ldots, Q_r]$, where $Q_i \in \mathbb{R}^{n\times n_i-1}$ corresponds to C_i. Moreover, each column sums to zero in the positions associated with C_i and has zeros in the other positions. In other words,

$$Q_i = \begin{bmatrix} \mathbf{0} \\ \tilde{Q}_i \\ \mathbf{0} \end{bmatrix}_{n\times(n_i-1).}$$

In Q_i, the upper and lower parts are zero matrices with appropriate dimensions (possibly empty). One possible choice of $\tilde{Q}_i$ would be

$$\tilde{Q}_i = \begin{bmatrix} I_{n_i-1} \\ -\mathbf{1}^T \end{bmatrix}_{n_i \times (n_i-1)}, \tag{54}$$

where $\mathbf{1} \in \mathbb{R}^{n_i-1}$ is a vector with ones in each position. Based on this method, the Q matrix for the equitable partition in Figure 22(a) can be given by $Q = Q_2 = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \end{bmatrix}^T$, and thus, $\tilde{Q}_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$.

The other choice of Q matrix is using the orthonormal basis of $\mathcal{R}(P)^\perp$. We denote this matrix as $\bar{Q}$. If we define

$$\bar{P} = P(P^TP)^{-\frac{1}{2}}. \tag{55}$$

Note that the invertibility of P^TP follows from the fact that the cells of the partition are nonempty [5]. Moreover, it satisfies that $\bar{P}^T\bar{Q} = \mathbf{0}$ and $\bar{Q}^T\bar{Q} = I_{n-r}$. In other words,

$$T = [\bar{P} \mid \bar{Q}] \tag{56}$$

is a matrix whose columns are defined on an orthonormal basis of $\mathbb{R}^n$ based on the equitable partition π, and $\bar{P}$ and $\bar{Q}$ have the same column spaces as P and Q respectively.

Theorem 6.3.8. *([57] Theorem 9.3.3) If π is an equitable partition of a graph $\mathcal{G}$, then the characteristic polynomial of $\hat{\mathcal{A}} = \mathcal{A}(\mathcal{G}/\pi)$ divides the characteristic polynomial of $\mathcal{A}(\mathcal{G})$.*

Lemma 6.3.9. *([57] Theorem 9.5.1) Let $\Phi \in \mathbb{R}^{n\times n}$ be a real symmetric matrix and let $R \in \mathbb{R}^{n\times m}$ be such that $R^TR = I_m$. Set $\Psi = R^T\Phi R$ and let $v_1, v_2, \ldots, v_m$ be an orthogonal set of eigenvectors for Ψ such that $\Psi v_i = \theta_i(\Psi)v_i$, where $\theta_i(\Psi) \in \mathbb{R}$ is an eigenvalue of Ψ. Then*

[5] In fact, P^TP is a diagonal matrix with $(P^TP)_{ii} = |C_i|$.

1. *The eigenvalues of* Ψ *interlace the eigenvalues of* Φ.

2. *If* $\theta_i(\Psi) = \theta_i(\Phi)$ *then there is an eigenvector* v *of* Ψ *with eigenvalue* $\theta_i(\Psi)$ *such that* Rv *is an eigenvector of* Φ *with eigenvalue* $\theta_i(\Phi)$.

3. *If* $\theta_i(\Psi) = \theta_i(\Phi)$ *for* $i = i, \ldots, l$, *then* Rv_i *is an eigenvector for* A *with eigenvalue* $\theta_i(\Phi)$ *for* $i = i, \ldots, l$.

4. *If the interlacing is tight, then* $\Phi R = R\Psi$.

Now we are in the position to prove the sufficiency of Lemma 6.3.1, i.e., if $\mathcal{L}$ and $\mathcal{L}_f$ share a common eigenvalue, the system $(-\mathcal{L}_f, -l_{fl})$ is not completely controllable.

Proof of Sufficiency of Lemma 6.3.1: Since $\mathcal{L}_f$ is a principle sub-matrix of $\mathcal{L}$, it can be given by

$$\mathcal{L}_f = R^T \mathcal{L} R,$$

where $R = [I_{n_f}, \mathbf{0}]^T \in \mathbb{R}^{n \times n_f}$. Following Lemma 6.3.9(2), if $\mathcal{L}_f$ and $\mathcal{L}$ share a common eigenvalue, say λ, then the corresponding eigenvector satisfies

$$v = Rv_f = \begin{bmatrix} v_f \\ \mathbf{0} \end{bmatrix},$$

where v is λ's eigenvector of $\mathcal{L}$ and v_f is that of $\mathcal{L}_f$. Moreover, we know that

$$\mathcal{L}v = \begin{bmatrix} \mathcal{L}_f & l_{fl} \\ l_{fl}^T & \mathcal{L}_l \end{bmatrix} \begin{bmatrix} v_f \\ \mathbf{0} \end{bmatrix} = \lambda \begin{bmatrix} v_f \\ \mathbf{0} \end{bmatrix},$$

which gives us $l_{fl}^T v_f = \mathbf{0}$, and thus the system is uncontrollable. ∎

Now, we have shown that the existence of a common eigenvalue shared by $\mathcal{L}$ and $\mathcal{L}_f$ is a necessary and sufficient condition for the leader-follower network to be uncontrollable.

6.3.3 A Necessary Condition

The way we approach this necessary condition is through Lemma 6.3.1. In what follows we will show first that both $\mathcal{L}$ and $\mathcal{L}_f$ are both similarity to some block diagonal matrices. Then we will, furthermore, show that under some circumstances, the diagonal block matrices, resulted from diagonalize $\mathcal{L}$ and $\mathcal{L}_f$, have some diagonal block(s) in common.

Lemma 6.3.10. *If a graph $\mathcal{G}$ has a nontrivial equitable partition (NEP) π with characteristic matrix P, the adjacency matrix $\mathcal{A}(\mathcal{G})$ of the graph is similar to a diagonal matrix*

$$\bar{\mathcal{A}} = \begin{bmatrix} \mathcal{A}_P & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_Q \end{bmatrix},$$

where $\mathcal{A}_P$ is similar to the adjacency matrix $\hat{\mathcal{A}} = \mathcal{A}(\mathcal{G}/\pi)$ of the quotient.

Proof. Let the matrix $T = [\bar{P} \mid \bar{Q}]$ be the orthonormal matrix with respect to π, as what we have defined in (56).

Now, let

$$\bar{\mathcal{A}} = T^T \mathcal{A} T = \begin{bmatrix} \bar{P}^T \mathcal{A} \bar{P} & \bar{P}^T \mathcal{A} \bar{Q} \\ \bar{Q}^T \mathcal{A} \bar{P} & \bar{Q}^T \mathcal{A} \bar{Q} \end{bmatrix}. \tag{57}$$

Since $\bar{P}$ and $\bar{Q}$ have the same column spaces as P and Q respectively, they inherit the $\mathcal{A}-$invariance property, i.e.,

$$\mathcal{A}\bar{P} = \bar{P}B \quad \text{and} \quad \mathcal{A}\bar{Q} = \bar{Q}C.$$

for some matrices B and C. Since their column spaces are orthogonal complements to each other, we get

$$\bar{P}^T \mathcal{A} \bar{Q} = \bar{P}^T \bar{Q} C = \mathbf{0}$$

and

$$\bar{Q}^T \mathcal{A} \bar{P} = \bar{Q}^T \bar{P} B = \mathbf{0}.$$

In addition, let $D_p^2 = P^T P$, we get

$$\begin{aligned} \bar{P}^T \mathcal{A} \bar{P} &= D_P^{-1} P^T \mathcal{A} P D_P^{-1} \\ &= D_P (D_P^{-2} P^T \mathcal{A} P) D_P^{-1} \\ &= D_P \hat{\mathcal{A}} D_P^{-1}, \end{aligned} \tag{58}$$

and therefore the first diagonal block is similar to $\hat{\mathcal{A}}$. ∎

Lemma 6.3.11. *Let P be the characteristic matrix of a NEP in $\mathcal{G}$. $\mathcal{R}(P)$ is K-invariant, where K is any diagonal block matrix of the form*

$$K = \operatorname{diag}(k_i I_{n_i})_{i=1}^r,$$

where $k_i \in \mathbb{R}$, $n_i = |C_i|$ is the cardinality of the cell, and $r = |\pi|$ is the cardinality of the partition. Consequently,

$$\bar{Q}^T K \bar{P} = \mathbf{0},$$

where $\bar{P} = P(P^T P)^{-\frac{1}{2}}$ and $\bar{Q}$ is chosen in such a way that $T = [\bar{P} \mid \bar{Q}]$ is a orthonormal matrix.

Proof. Since

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_r \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & \cdots & p_r \end{bmatrix},$$

where $P_i \in \mathbb{R}^{n_i \times r}$ is a row block which has 1's in column i and 0's elsewhere. On the other hand p_i is a characteristic vector representing C_i, which has 1's in the positions associated with C_i and zeros otherwise. For instance, the example given in (52) can

be written as

$$P = \left[\begin{array}{c|c|cc} 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \tag{59}$$

and we can find

$$P_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

while $p_2 = [0\ 1\ 1\ 0\ 0]^T$.

With a little bit calculation we can find

$$KP = \begin{bmatrix} k_1 P_1 \\ k_2 P_2 \\ \vdots \\ k_r P_r \end{bmatrix} = \begin{bmatrix} k_1 p_1 & k_2 p_2 & \dots & k_r p_r \end{bmatrix} = P\hat{K},$$

where $\hat{K} = \mathrm{diag}(k_i)_{i=1}^r$, which shows that $\mathcal{R}(P)$ is K-invariant. Since $\mathcal{R}(\bar{Q}) = \mathcal{R}(P)^\perp$, it is K-invariant as well by Lemma 6.3.7, and

$$\bar{Q}^T K \bar{P} = \bar{Q}^T \bar{P} \hat{K} = \mathbf{0}.$$

The second equation follows the fact that $\mathcal{R}(\bar{Q})$ and $\mathcal{R}(\bar{P})$ are orthogonal complement to each other. ∎

By the definition of the equitable partition, the subgraph induced by a cell is regular and every node in the same cell has the same number of neighbors outside the cell. Therefore, the nodes belonging to the same cell have the same degree, and thus by Lemma 6.3.11, $\mathcal{R}(\bar{Q})$ and $\mathcal{R}(P)$ are $\mathcal{D}$-invariant, where $\mathcal{D}$ is the degree matrix given by

$$\mathcal{D} = \mathrm{diag}(d_i I_{n_i})_{i=1}^r,$$

where $d_i \in \mathbb{R}$ denotes the degree of each nodes in cell.

Since the graph Laplacian satisfy $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$, Lemma 6.3.10 and Lemma 6.3.11 together can show that $\mathcal{R}(\bar{Q})$ and $\mathcal{R}(P)$ are $\mathcal{L}$-invariant, and thus, we have following corollary

Corollary 6.3.12. *Given the same condition as in Lemma 6.3.10 $\mathcal{L}$ is similar to a diagonal block matrix*

$$\bar{\mathcal{L}} = T^T \mathcal{L} T = \begin{bmatrix} \mathcal{L}_P & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_Q \end{bmatrix}, \tag{60}$$

where $\mathcal{L}_P = \bar{P}^T \mathcal{L} \bar{P}$ and $\mathcal{L}_Q = \bar{Q}^T \mathcal{L} \bar{Q}$, and $T = [\bar{P} \mid \bar{Q}]$ defines a orthonormal basis for $\mathbb{R}^n$ with respect to π.

As (60) defines a similarity transformation, it follows that $\mathcal{L}_P$ and $\mathcal{L}_Q$ carry all the spectrum information of $\mathcal{L}$, i.e., they share eigenvalues with $\mathcal{L}$.

Now that, as we have show in (46), in a leader-follower network, the graph Laplacian can be partitioned as

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_f & l_{fl} \\ l_{fl}^T & \mathcal{L}_l \end{bmatrix}$$

according to the leader assigning scheme. Transformations similar to (60) can be found for $\mathcal{L}_f$ in the presence of NEPs in the follower graph $\mathcal{G}_f$.

Corollary 6.3.13. *Let $\mathcal{G}_f$ be a follower graph, and let $\mathcal{L}_f$ be the diagonal sub-matrix of $\mathcal{L}$ related to $\mathcal{G}_f$. If there is a NEP π_f in $\mathcal{G}_f$ and a π in $\mathcal{G}$, such that all the nontrivial cells in π_f are also cells in π, there exists an orthonormal matrix T_f such that*

$$\bar{\mathcal{L}}_f = T_f^T \mathcal{L}_f T_f = \begin{bmatrix} \mathcal{L}_{fP} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_{fQ} \end{bmatrix}. \tag{61}$$

Proof. Let $\bar{P}_f = P_f (P_f^T P_f)^{-\frac{1}{2}}$, where P_f is the characteristic matrix for π_f, and let $\bar{Q}_f$ be defined on a orthonormal basis of $\mathcal{R}(P_f)^{\perp}$. In the above way, we have obtained

an orthonormal basis for $\mathbb{R}^{n_f}$ with respect to π_f. Moreover, $\mathcal{L}_f = \mathcal{D}_f - \mathcal{A}_f$ where $\mathcal{A}_f$ denotes the adjacency matrix of $\mathcal{G}_f$ while $\mathcal{D}_f$ is the degree matrix corresponding to the original graph $\mathcal{G}$. Since all the nontrivial cells in π_f are also cells in π, $\mathcal{D}_f$ satisfies the condition in Lemma 6.3.11, i.e., nodes from an identical cell in π_f have the same degree. Hence from Lemma 6.3.10 and Lemma 6.3.11, $\mathcal{R}(P)$ and $\mathcal{R}(\bar{Q})$ are $\mathcal{L}_f$-invariant and thus

$$\bar{\mathcal{L}}_f = T_f^T \mathcal{L}_f T_f = \begin{bmatrix} \mathcal{L}_{fP} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_{fQ} \end{bmatrix}, \tag{62}$$

where $T_f = [\bar{P}_f \mid \bar{Q}_f]$, $\mathcal{L}_{fP} = \bar{P}_f^T \mathcal{L}_f \bar{P}_f$ and $\mathcal{L}_{fQ} = \bar{Q}_f^T \mathcal{L}_f \bar{Q}_f$, . ∎

Again, the diagonal blocks of $\bar{\mathcal{L}}_f$ share all the spectrum information with $\mathcal{L}_f$. Now, we are in the position to prove our main result.

Theorem 6.3.14. *Given a connected graph $\mathcal{G}$ and the induced follower graph $\mathcal{G}_f$, the system $(-\mathcal{L}_f, -l_{fl})$ is not complete controllable if there exist NEPs on $\mathcal{G}$ and $\mathcal{G}_f$, say π and π_f, such that all the nontrivial cells of π are contained in π_f, i.e., $\exists\ \pi$ and π_f, such that $|C_i| = 1,\ \forall\ C_i \in \pi \backslash \pi_f$.*

Proof. In Corollary 6.3.12 and Corollary 6.3.13, we have already shown that $\mathcal{L}$ and $\mathcal{L}_f$ are both similar to some diagonal block matrices. Here we want to show the relation ship between these diagonal block matrices.

Assume $\pi \cap \pi_f = \{C_1, C_2, \ldots, C_{r_1}\}$. According to the given condition, $|C_i| \geq 2,\ i = 1, 2, \ldots, r_1$. Without loss of generality, we can index the nodes in such a way that the nontrivial cells comprise the first n_1 nodes [6] such that

$$n_1 = \sum_{i=1}^{r_1} |C_i| \leq n_f < n.$$

[6] We introduce n_1 for convenience. It is easy to verify that $n_1 - r_1 = n - r = n_f - r_f$

Since all the nontrivial cells of π are in π_f, their characteristic matrices have similar structures

$$P = \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & I_{n-n_1} \end{bmatrix}_{n \times r} \quad \text{and} \quad P_f = \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & I_{n_f-n_1} \end{bmatrix}_{n_f \times r_f},$$

where P_1 is a $n_1 \times r_1$ matrix that contains the nontrivial part of the characteristic matrices. Since $\bar{P}$ and $\bar{P}_f$ are only normalized P and P_f respectively, they have the same block structures. Consequently $\bar{Q}$ and $\bar{Q}_f$, the matrices containing orthonormal basis of $\mathcal{R}(P)$ and $\mathcal{R}(P_f)$, have following structures

$$\bar{Q} = \begin{bmatrix} Q_1 \\ \mathbf{0} \end{bmatrix}_{n \times (n_1 - r_1)} \quad \text{and} \quad \bar{Q}_f = \begin{bmatrix} Q_1 \\ \mathbf{0} \end{bmatrix}_{n_f \times (n_1 - r_1)},$$

where Q_1 is a $n_1 \times (n_1 - r_1)$ matrix that satisfies

$$Q_1^T P_1 = \mathbf{0}.$$

As one can observe, $\bar{Q}_f$ is different from $\bar{Q}$ only by $n - n_f$ rows of zeros. In other words, the special structures of $\bar{Q}$ and $\bar{Q}_f$ gives us the relationship

$$Q_f = R^T Q,$$

where $R = [I_{n_f}, 0]^T$.

Now, recall the definition of $\mathcal{L}_Q$ and $\mathcal{L}_{fQ}$ from (60) and (62), which gives us

$$\begin{aligned} \mathcal{L}_Q &= \bar{Q}^T \mathcal{L} \bar{Q} \\ &= \bar{Q}_f^T R^T \mathcal{L} R \bar{Q}_f \\ &= \bar{Q}_f^T \mathcal{L}_f \bar{Q}_f = \mathcal{L}_{fQ}. \end{aligned} \tag{63}$$

Therefore $\mathcal{L}_f$ and $\mathcal{L}$ share the same eigenvalues associated with $\mathcal{L}_Q$, and by Lemma 6.3.1, the system is not completely controllable. ∎

Next, we further show that actually T_f is a similarity transformation leads us to the controllability transformation.

Theorem 6.3.15. *Given a leader-follower network satisfies the condition in Theorem 6.3.14, T_f is a similarity transformation that gives us the controllability decomposition of the system.*

$$\dot{x}_f \;\; = \;\; -\mathcal{L}_f x_f - l_{fl} x_l, \tag{64}$$

.

Proof. Assume there is a NEP π_f such that $|\pi_f| = r < n_f$. Let P_f be the characteristic matrix corresponding to the equitable partition π in $\mathcal{G}_f$. As shown in Corollary 6.3.13 $\mathcal{L}_f$ is similar to a diagonal block matrix $\bar{\mathcal{L}}_f$. What remains to be shown is that the same coordinate transformation on l_{fl} will result in a block structure such that the last $(n_f - r)$ elements are zeros, i.e.,

$$T_f^T l_{fl} = \bar{l}_{fl} = \begin{bmatrix} l_c \\ \\ \mathbf{0} \end{bmatrix}, \tag{65}$$

where $l_c = \bar{P}_f^T l_{fl} \in \mathbb{R}^r$ and $T_f = [\bar{P}_f \mid \bar{Q}_f]$. In other words, we need to prove that $\bar{Q}_f^T l_{fl} = 0$. We can show this by directly observation. On the one hand, recall the properties of matrix Q introduced in Lemma 6.3.6, which shows that the columns of Q have nonzero elements only in the position associated with nontrivial cells. Moreover, they sum to zero in those positions. Since $\bar{Q}_f$ has same column space as Q_f, its columns has the same structure with respect to the nontrivial cells in π_f. On the other hand, each the columns of l_{fl} denote how the followers are connected to a leader. Column i of l_{fl} would have a -1 in j-th element if there is an edge between leader i and follower j. Since, under π, all the nontrivial cells are in $\mathcal{G}_f$, all the leader nodes are trivial partitions, i.e., each leader is either connected to all the nodes in $C_i,\ i = 1, 2, \ldots, r_1$, or none. Thus columns of l_{fl} are always orthogonal to those of $\bar{Q}_f$, and hence $\bar{Q}_f^T l_{fl} = \mathbf{0}$. This shows that T_f gives us a valid similarity transformation for the controllability decomposition:

$$\dot{z}_f = -\bar{\mathcal{L}}_f z_f - \bar{l}_{fl} x_l, \tag{66}$$

where

$$z_f = T_f^T x_f$$

is the transformed state vector. Furthermore we can state that

$$\begin{aligned} \dot{z}_f^c &= -\mathcal{L}_{fP} z_f^c - l_c x_l && (67) \\ \dot{z}_f^{uc} &= -\mathcal{L}_{fQ} z_f^{uc}, && (68) \\ & && (69) \end{aligned}$$

where $z_f^c = \bar{P}_f^T x_f$ is the controllable part of the transformed state, while $z_f^{uc} = \bar{Q}_f^T x_f$ is the uncontrollable part of the transformed state. ∎

Remark 6.3.2. Recall that $\bar{P}_f = P_f(P_f^T P_f)^{-\frac{1}{2}}$ and $\mathcal{L}_{fP} = \bar{P}_f^T \mathcal{L}_f \bar{P}_f$. Multiply both side of (68) by $(P_f^T P_f)^{-\frac{1}{2}}$, we get

$$(P_f^T P_f)^{-1} P_f^T \dot{x}_f = (P_f^T P_f)^{-1} P_f^T \mathcal{L}_f P_f (P_f^T P_f)^{-1} P_f^T x_f + (P_f^T P_f)^{-1} P_f^T l_{fl} x_l. \quad (70)$$

Moreover, if we define $P_f^+ = (P_f^T P_f)^{-1} P_f^T$ as the pseudo inverse of P_f, and let $x_f^\pi = P_f^+ x_f$, (70) can be further simplified

$$\begin{aligned} \dot{x}_f^\pi &= P_f^+ \mathcal{L}_f P_f P_f^+ x_f + P_f^+ l_{fl} x_l && (71) \\ &= \hat{\mathcal{L}}_f P_f^+ x_f + P_f^+ l_{fl} x_l \\ &= \hat{\mathcal{L}}_f x_f^\pi + P_f^+ l_{fl} x_l, \end{aligned}$$

where $\hat{\mathcal{L}}_f = \mathcal{L}(\mathcal{G}/\pi) = P_f^+ \mathcal{L}_f P_f$ is the *generalized Laplacian matrix* of the quotient graph. It is related to $\bar{\mathcal{L}}_{fP}$ by the relation $\bar{\mathcal{L}}_{fP} = (P_f^T P_f)^{\frac{1}{2}} \hat{\mathcal{L}}_f (P_f^T P_f)^{\frac{1}{2}}$.

In the situation described in Theorem 6.3.14, the system is not completely controllable. This theorem thus gives us a method to identify uncontrollable situations in a leader-follower system. Intuitively speaking, vertices in the same cell of a NEP that satisfy the condition in Theorem 6.3.14 is not distinguishable from the leaders' point of view. In other words, if the agents belong to the same cell shared by π and

π_f and they start from the same point, it is impossible for the leaders to pull them apart. Thus the controllable subspace can be obtained by collapsing all the nodes in the same cell into a single meta-node. However, since the NEPs may not be unique, as we have seen in the case of the Peterson graph, more work is required before a complete understanding of this issue is obtained.

Note that our sufficient condition for a graph to be uncontrollable immediately produces a necessary condition for a graph to be controllable and this states as a corollary.

Corollary 6.3.16. *Given a connected graph $\mathcal{G}$ with the induced follower graph $\mathcal{G}_f$, a necessary for $(-\mathcal{L}_f, -l_{fl})$ to be controllable is that no NEPs π and π_f on $\mathcal{G}$ and $\mathcal{G}_f$ exist such that π and π_f share all nontrivial cells.*

Corollary 6.3.17. *If $\mathcal{G}$ is disconnected, a necessary condition for $(-\mathcal{L}_f, -l_{fl})$ to be controllable is that all of its connected components are controllable.*

6.4 Examples and Discussions

In this part, we will show some uncontrollable situations that are identifiable by our method, and discuss the relationship among some existing results, our result and our ultimate goal.

a) Single Leader with Symmetric Followers. In Figure 22, if we choose node '5' as the leader, the symmetric pair (2,3) in the follower graph renders the network uncontrollable as stated in [43] . The dimension of the controllable subspace is three, while there are four nodes in the follower group. This result can also be interpreted by Theorem 6.3.14, since all the automorphism groups introduce equitable partitions.

b) Single Leader with Equal Distance Partitions. We have shown in Figure 21 that the Peterson graph has two NEPs. One is introduced by the automorphism groups and the other (π_2) is introduced by the equal distance groups. Based on π_2, if we choose node '1' as the leader, the leader-follower group ends up with a controllable subspace

with dimension of two. Since there are four orbits[7] in the automorphism groups, this dimension can only be interpreted by the two-cell equal distance partitions[8].

c) Multiple Leaders. The last example is a modified leader graph based on the peterson graph. In Figure 23, we add another node ('11') connected to $\{3, 4, 7, 8, 9, 10\}$ as the second leader in addition to node '1'. In this network, there is an equal distance partition with four cells, $\{1\}$, $\{2, 5, 6\}$ $\{3, 4, 7, 8, 9, 10\}$ and $\{11\}$. In this situation, the dimension of the controllable subspace is still two, which is consistent with example b).

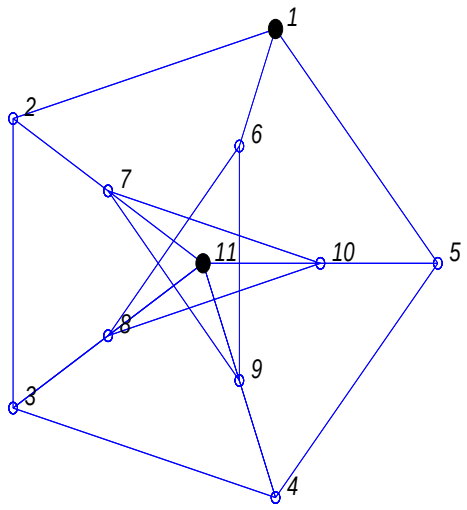


Figure 23. A 2-leader network based on the Peterson graph. The second leader '11' is connected to '3','4','7','8','9' and '10'.

6.5 Conclusions

In this chapter, we considered the control of the leader-follower structure over a heterogeneous network. We first derived a set of transformations that can be employed to derive the system matrices for the case where one or more of the nodes (leader nodes)

[7]They are $\{2, 5, 6\}, \{7, 10\}, \{8, 9\}, \{3, 4\}$
[8]They are $\{2, 5, 6\}$ and $\{3, 4, 7, 8, 9, 10\}$

update their state values based on an external command. The other nodes (followers) are assumed to update their states according to their relative states with their neighbors. In such a setting, we studied the controllability of the resulting dynamic system. It was shown that there is a intricate relationship between uncontrollability of the system and various graph theoretic properties of the network. In particular, we pointed out the importance of the nontrivial equitable partitions of a network in the controllability properties of the interconnected system. The results of this chapter belong to the intersection of graph theory and control theory and contribute to the study of system-theoretic issues from a purely graph-theoretic outlook.

One important application of the leader-follower structure is formation control. Ideally, the leader can move in such a way that the followers will be lead into desired locations. In order for the followers to achieve their desired positions in finite time, we need to employ optimal control techniques, which is shown as an example in Section 9.2.

CHAPTER 7

DISTRIBUTED ESTIMATION IN HETEROGENEOUS NETWORKED SYSTEMS

7.1 Introduction

Distributed sensor networks (DSNs) are often used to monitor spatially distributed parameters. Each sensor node is embedded with a microprocessor for simple information processing, and a transceiver for information exchange. Besides the sensor nodes, one or more central nodes are connected to the network to collect data and make decisions. These central nodes are more advanced than the sensor nodes in their communication and processing capabilities.

DSNs are flexible platforms for distributed sensing and information processing. However, as we have discussed in Section 1.2.2.2, the existing methods, namely multi-hop and mobile central nodes, have their own shortcomings. Heuristics can be learned from the distributed multi-agent system, where a global behavior is achievable even when each individual agent makes its own decision solely based on the local information. The agents exchange and process local information in such a coordinated way that the information is quickly propagated through the network. Having a certain amount of processing, communication, and storage capability, the sensor node should also be able to disseminate the information based on a consensus-like distributed rule.

This chapter proposes an estimation scheme which, instead of monitoring the value of each sensor node, just observes the states of a subset of the nodes, and uses this information to recover the spatial distribution of the parameter. Ideally, only a small portion of the nodes is needed to recover the distribution. This chapter is arranged as follows: In Section 7.2, we first introduce a graph-based model for DSN and then propose a distributed estimation strategy based on the rendezvous algorithm. In Section 7.3 we study the observability of the DSN and give a necessary condition.

7.2 Graph-Based Estimation of Distributed Sensor Networks

Given a DSN with N sensors, the underlying *sensor graph*, $\mathcal{G}_s = (V_s, E_s)$, takes the sensors as its nodes, and the communication links between node pairs as its edges. Without lose of generality, let $p_i \in \mathbb{R}$ be the sensor value of node $i,\ i = 1, \ldots, N$. Next, let $y \in \mathbb{R}^N$ be the state of a distributed estimator, and y_i be the component associated with node i. Note that y_i is different form its real sensed value p_i in that y_i decided by a distributed estimation algorithm. It is y_is, not p_is, that are transmitted through the communication links and computed at the processors embedded with the sensor nodes. In other words, each node has three roles: First, it is a sensor that monitors the parameter value, p_i. Secondly, it is a processing unit of a distributed computing system, which updates its state, y_i, by processing the information gathered from its communication links. Thirdly, it is a transceiver that exchange its state information with its neighbors $(y_j,\ j \in \mathcal{N}(i))$.

Our desired estimation algorithm can then be described in the following way. At time zero, node i records its sensing value p_i and use it as the initial value for y_i. Then it updates its own state based on the local information, i.e., all the states of its neighbors and the state of itself. The algorithm can be formulated as

$$\frac{dy_i}{dt} = \sum_{j \in \mathcal{N}(i)} f(y_i(t) - y_j(t)),\ y_i(0) = p_i,\ i = 1, \ldots, N, \tag{72}$$

where, again, $\mathcal{N}(i)$ is the neighborhood of node i. Based on this updating scheme, the central nodes retrieve observation

$$z(t) = g(y(t)), \tag{73}$$

where $y(t) = [y_1(t), \ldots, y_N(t)]^T$, and $z(t) \in \mathbb{R}^m,\ m << N$, with m being the number of the central nodes. Our goal here is to uniquely determine $p = [p_1, \ldots, p_N]^T$ from $z(t)$.

If we adopt a nearest neighbor rule for the $f(\cdot)$ in (72), we get

$$\frac{dy_i}{dt} = \sum_{j \in \mathcal{N}(i)} (y_j - y_i),\ i = 1, 2, \ldots, N. \tag{74}$$

Using the graph-based control notation, (74) can be written as

$$\begin{cases} \dot{y} &= -\mathcal{L}(\mathcal{G}_s)y, \quad y(0) = p, \\ z &= \quad Cy, \end{cases} \tag{75}$$

where $\mathcal{L}(\mathcal{G}_s)$ is the Laplacian matrix of the graph associated with the sensor network, and $C \in \mathcal{R}^{m \times N}$ is the observation matrix. Since the graph is assumed to be static, we will use $\mathcal{L}_s$ for $\mathcal{L}(\mathcal{G}_s)$ wherever it causes no confusion. As we have proved in Chapter 3 that system (75) is stable, and $y_i(t)$ converges to $\sum_1^N p_j/N$, for all $i \in \{1, N\}$. Nonetheless, from control theory, we know that if the system $(-\mathcal{L}_s, C)$ is observable, we can fully recover the initial state $y(0)$ from the observation $z(t)$. Note that, in the case of multiple central nodes, each central node collects raw information from a subset of sensors and sums them up together, an operation that equals to multiply y by each row of C. Then the summation results are reported to a single super node in order for final information processing and decision making.

We further restrict C to be a $(0, 1)$ matrix, meaning that all the links are equally weighted. If C has a nonzero element in column j, row i, it means that there is a communication link between sensor j and central node i.

One can design different observer based on (75), and a simple way of recovering the information, assuming the parameter being static or changing very slowly, is to solve the system

$$z(t) = Ce^{-t\mathcal{L}_s}y(0),\ t \geq 0,$$

or in the discrete time form

$$\begin{bmatrix} z(0) \\ z(1) \\ \vdots \\ z(T-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ -C\mathcal{L}_s \\ \vdots \\ C(-\mathcal{L}_s)^{T-1} \end{bmatrix}}_{\mathcal{O}_T} y(0),$$

where $\mathcal{O}_T$ becomes the observability matrix when $T = N$. Of course, more sophisticated estimation methods are available in the literature, but it is beyond the scope of this thesis.

As we can see, the proposed estimation scheme is relying on one crucial assumption: the system $(-\mathcal{L}_s, C)$ is observable. Now, let us revisit the observability theorem from control theory.

Lemma 7.2.1. *For a LTI system*

$$\begin{cases} \dot{x} &= Ax, \quad x(0) = x_0 \\ y &= Cx, \end{cases} \tag{76}$$

the following are equal:

- *system (76) is observable*
- *the rank of the observability matrix* $(\mathcal{O}(N))$ *is* N
- *rank of* $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ *equals* n *for every eigenvalue* λ *of* A

We can always check the rank condition to determine the observability of the network. However, it becomes infeasible when the number of nodes becomes very large. We need to guarantee the observability when we build the network, and, for this sake, we want to understand how the topology of the network affects the observability. In the following section, we will show how the observability property is related to the topology of the sensor network.

7.3 Observability Analysis of Distributed Sensor Networks

The results shown in this section are closely related to the results from Chapter 6 because of the duality between the observability problem and the controllability problem. Let us recall some notations related to the nontrivial equitable partition (NEP). Given a DSN, suppose its underlying sensor graph has a NEP π_s with $|\pi_s| = r$, and let P_s be the characteristic matrix of π_s. With a little abuse of notation, we further let $\bar{P}_s = P_s(P_s^T P_s)^{-\frac{1}{2}}$ be the normalized characteristic matrix of P_s and $\bar{Q}$ be chosen in such a way that renders the matrix $T_s = [\bar{P}_s \mid \bar{Q}_s]$ an orthonormal matrix.

First, consider the single central node case and denote the central node by V_{N+1}. Now, C is a row vector and z is a scalar equal to the sum of the states of the nodes connecting to the central node. Since C is a $(0, 1)$ matrix, it can not belong to the orthogonal complement of P. If it happens to be the character vector of a non-trivial cell, some states will become unobservable.

Theorem 7.3.1. *Let $\mathcal{G}$ be the underlying graph of a DSN and π be the only one NEP over $\mathcal{G}_s$. System* (75) *is not complete observable if the central node connects to all the nodes in one or more cells, i.e.,*

$$\mathcal{N}(N+1) = \{\cup_{l=1}^{k} C_{i_l}\},\ k, i_l \in \{1, \ldots, r\}. \tag{77}$$

Proof. The proof follows directly from Corollary 6.3.12, Theorem 6.3.14 and Theorem 6.3.15. ∎

Corollary 7.3.2. *Let $\mathcal{G}_s$ be the underlying graph of a DSN. If* (77) *is valid for all the NEPs over $\mathcal{G}_s$. System* (75) *is not completely observable.*

In the case of multiple central nodes, simply apply Theorem 7.3.1 to each leader will not lead us to the right conclusion. From the duality, we know that the system given in (76) is observable if and only if (A^T, C^T) is controllable. Thus, we want to utilize the results in Chapter 6, especially Theorem 6.3.14 and Theorem 6.3.15.

Given a DSN, let us regard the central nodes as the leaders in the leader-follower structure, and denote the resulting graph as the *augmented sensor graph* $\mathcal{G}$. $\mathcal{G}$ is nothing more than adding the central nodes to the sensor graph $\mathcal{G}_s$ and adding edges onto $\mathcal{G}_s$, which represents the information flows from sensor nodes to the central nodes. The sensor graph $\mathcal{G}_s$ plays the similar role of the follower graph, $\mathcal{G}_f$, in the leader-follower structure, as shown in Figure 24. The difference is that in the DSN the information flows from the sensor nodes to the central nodes, while in the leader-following structure, the information flows from the leaders to the followers. With this construction, we are ready to state our main theorem.

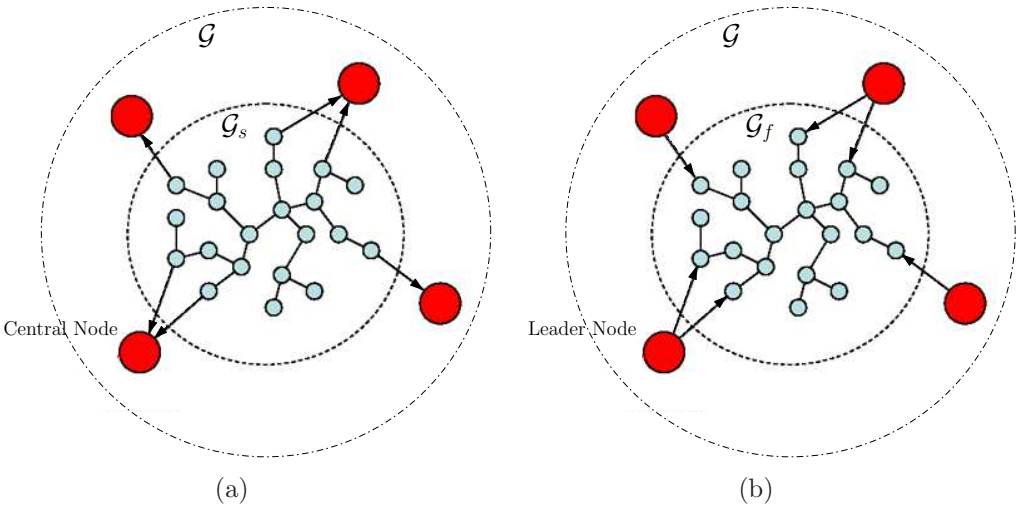


Figure 24. Shown in the picture are (a) the graph associate with a DSN ($\mathcal{G}_s$), and the augmented graph $\mathcal{G}$, and (b) the interaction graph $\mathcal{G}$ of the leader-follower structure with the same topology as the DSN in (a). Note the follower graph $\mathcal{G}_f$ is similar to $\mathcal{G}_s$, but the information flows are from the leader to the follower.

Theorem 7.3.3. *Given a DSN with connected graph $\mathcal{G}_s$ and the augmented graph $\mathcal{G}$, the system* (75) *is not complete observable if there exist NEPs on $\mathcal{G}$ and $\mathcal{G}_s$, say π and π_s, such that all the nontrivial cells of π are contained in π_s, i.e., $\exists$ π and π_s, such that $|C_i| = 1$, $\forall$ $C_i \in \pi \backslash \pi_s$. Moreover, $T_s = [\bar{P}_s \mid \bar{Q}_s]$ is a similarity transformation that gives us the observability decomposition of the system.*

Proof. Let P_s be the characteristic matrix corresponding to the NEP π_s in $\mathcal{G}_s$. As

shown in Corollary 6.3.12

$$\mathcal{L}_s \sim \bar{\mathcal{L}}_s = \begin{bmatrix} \mathcal{L}_{sP} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_{sQ} \end{bmatrix}$$

as a result of the similarity transformation T_s. Next, we want to show that the same transformation on C can give us desired block structure. In other words, we want to show that

$$CT_s = \bar{C} = [C_o, \mathbf{0}]. \tag{78}$$

From Lemma 6.3.6, we know that $\bar{Q}_s$ have zero column sum and that the columns could have nonzero elements only in the rows associated with nontrivial cells. On the other hand, row i of C denotes how the sensor nodes are connected to central node i, thus it would have a 1 in j-th element if there is an edge between central node i and sensor node j. Since, under π, all the nontrivial cells are in $\mathcal{G}_s$, and all the central nodes are trivial partitions, i.e., each leader is either connected to all the nodes in C_i, for some $i \in 1, 2, \ldots, r_1$, or none. Thus columns of C^T are in the column space of P_s. Therefore, we have

$$C\bar{Q}_s = \mathbf{0},$$

and the transformed system can be given as

$$\begin{cases} \dot{v} &= -\bar{\mathcal{L}}_s \, v, \quad v(0) = T_s^T p, \\ z &= \bar{C} \, v, \end{cases} \tag{79}$$

where

$$v = T_s^T y$$

is the transformed state vector. Furthermore we can state that

$$\dot{v}^o = -\mathcal{L}_{sP} \, v^o \tag{80}$$

$$\dot{v}^{uo} = -\mathcal{L}_{sQ} \, v^{uo} \tag{81}$$

$$z = C_o v^o \tag{82}$$

where $v^o = \bar{P}_s^T y$ is the observable part of the transformed state, while $v^{uo} = \bar{Q}_s^T y$ is the unobservable part of the transformed state. ∎

Remark 7.3.1. If we let $y^\pi = P_s^+ y$, where $P_s^+ = (P_s^T P_s)^{-1} P_s^T$ is the pseudo inverse of P_s, and multiply both side of (81) by $(P_s^T P_s)^{-\frac{1}{2}}$ we get

$$\begin{aligned} \dot{y}^\pi &= P_s^+ \mathcal{L}_s P_s P_s^+ y \\ &= \hat{\mathcal{L}}_s P_s^+ y \\ &= \hat{\mathcal{L}}_s y^\pi, \end{aligned} \tag{83}$$

where $\hat{\mathcal{L}}_s$ is the Laplacian for the quotient $(\mathcal{G}_s/\pi)$.

To this end, we have derived the sufficient condition for a DSN not to be completely controllable. The converse of this theorem gives us an necessary condition for the DSN to be fully observable. We state this as a corollary

Corollary 7.3.4. *Given a connected sensor graph $\mathcal{G}_s$ and its augmented graph $\mathcal{G}$, a necessary condition for $(-\mathcal{L}_s, C)$ to be observable is that no NEPs π and π_s exist over $\mathcal{G}$ and $\mathcal{G}_s$ such that π and π_s share all nontrivial cells.*

7.4 Conclusions and Discussions

A novel estimation strategy for distributed sensor network was proposed in this chapter. In this strategy, each sensor node runs a rendezvous-like scheme on their local information set and the central nodes estimate the original information by observing a small set of the sensors. Observability, an crucial issue enabling our strategy, was discussed and a set of necessary conditions were given.

The merit of our method is three folds. First, it requires the minimum number of information exchanges between the sensor nodes and the central nodes. In a multi-hop strategy with only one central node, the network is essentially a star graph with the central node at its center. N packets of data, each containing the information from a

sensor node, arrive at the central node directly or through routing. Whereas, in our method, the number of packets received by the central node is only determined by the number of sensor nodes they connected to. If the network is correctly constructed, we need to read only one node to recover the whole scene[1].

Secondly, the amount of information exchanged between sensor nodes does not change with respect to the distance to the central node. In a n-hop route, there are $\sum_1^{n+1} i$ packets transmitted from the starting node to the central node. Furthermore, with the same number of central node, n increases dimensionally (quadratically in 2-D, and cubically in 3-D) as the field scales. With our strategy, however, every link need to carry only 2 packets of information, so with the same length n only $2n + 1$ packet are transmitted. (The central nodes only receive one packet from each sensor connected to them.) The minimum communication does not come without a cost. Each node has to sum up their neighbor's information and the central node has to infer the parameter field by solve a set of linear equations. However, concerning that computation nowadays is much cheaper and consumes much less power than communication[2], our method still holds great advantage over the multi-hop strategy. Although the mobile-central node need to collect only N packet in a complete coverage, comparing $2M + m$ in our method, M being the number of communication links which is normally larger than N, our method is much faster in a large field.

Thirdly, our method is less complex comparing to multi-hop or mobile central node strategy. Each sensor node simply calculates the average, without even know their neighbors' identity. It does not have to deal with the routing problem, a crucial part in the multi-hop. From the central nodes point of view, there is need to walk through the field to collect the data, so they do not have to worry about the coverage problem or motion planning.

[1]Given the network is observable from this node.

[2]Somebody argue that communication is 2000 times expensive than computation to process the same amount of information

Since the central nodes have long range communication ability, they can share information and keep a copy of $z(t)$ for each of them. In this way, each of them can play the role of decision maker. A further extension would be such that each node acts as the central node and use the history of his own state, say $y_i(t)$ to infer the state of the others. Though this scheme is elegant and robust in the sense of fault tolerance, we will not pursuit this direction any further.

CHAPTER 8

CONTAINMENT CONTROL OF HETEROGENEOUS NETWORKED SYSTEMS

8.1 Introduction

This chapter investigates a variation of the leader-follower structure, namely the so-called *containment* problem. The problem is to drive a collection of autonomous, mobile agents to a given target location while guaranteeing that their motion satisfies certain geometric constraints. These constraints are there to ensure that the agents are contained in a particular area during their transportation. Such issues arise for example when a collection of autonomous robots are to secure and then remove hazardous materials. This removal must be secure in the sense that the robots should not venture into populated areas or in other ways contaminate their surroundings.

Based on the leader-follower structure, we will let the follower agents move autonomously based on local, consensus-like interaction rules, commonly found in the literature under the banner of algebraic graph theory [62, 66, 67]. However, we will augment this control structure with the addition of leader-agents or anchor nodes [42]. These leaders are to define vertices in a convex polytope (the leader-polytope) and they are to move in such a way that the target area is reached while ensuring that the follower-agents stay in the convex polytope spanned by the leaders, as shown in Figure 25. As such, the followers movements are calculated in a decentralized manner according to a fixed interaction topology, while the leaders are assumed to be able to detect if any of the followers violate the containment property.

For the leaders, we will use a hybrid Stop-Go policy [68, 69], in which the leaders move according to a decentralized formation control strategy until the containment property is about to be violated. At this point, they stop and let the followers settle back into the leader-polytope before they start moving again. For such a strategy

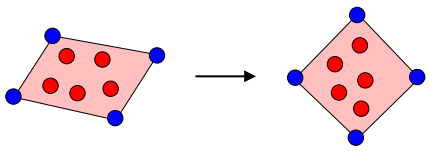


Figure 25. The containment problem: The leaders are to move in such a way that the followers remain in the convex leader-polytope for all times.

to be successful, a number of results are needed, including a guarantee that the Laplacian-based follower-control will in fact drive the followers back into the leader-polytope. Moreover, we must also ensure that such a control strategy is feasible in the sense of non-Zeno, live in the sense of not staying in the Stop mode indefinitely, and convergent in the sense that the target area is in fact reached.

After the leader-follower case has been settled, we will generalize this situation by introducing multiple layers in the network. This allows us to define complexity and performance measures in terms of the information flow and algebraic connectivity in the network, and relate these to the number of layers in the network. As such this work consists an extension of our previous work on leader-follower control [40] and hierarchial multi-layer control [41] of multi-agent system.

8.2 Operations on Graphs

In this section we will present the basic mathematical framework and some enabling results in networked systems.

Definition 8.2.1. Let $S = (V_S, E_S)$ be an undirected host graph and $V_{S'} \subset V_S$. The *subgraph* S' associated with $V_{S'}$ is the pair $(V_{S'}, E_{S'})$ where $E_{S'} = \{(x, y) \in E_S : x \in V_{S'}, y \in V_{S'}\}$.

Definition 8.2.1 allows basic operations in set theory to be extended to graphs.

Definition 8.2.2. Let S_1 and S_2 be to subgraphs of the graph S. Then, $S_1 \cup S_2$, $S_1 \cap S_2$, $S_1 \backslash S_2$ are the graph associated with $V_{S_1} \cup V_{S_2}$, $V_{S_1} \cap V_{S_2}$, and $V_{S_1} \backslash V_{S_2}$, respectively.

For our purposes, we will often use graphs with a boundary.

Definition 8.2.3. Let S be a subgraph of $\mathcal{G}$. The boundary of S is the subgraph $\partial S \subset \mathcal{G}$ associated with $V_{\partial S} \doteq \{y \in V_{\mathcal{G}} \setminus V_S \ : \ \exists\, x \in V_S \ : \ x \sim y\}$. The closure of S is $\bar{S} = \partial S \cup S$.

Note that the definition of the boundary of a graph depends upon the host graph $\mathcal{G}$. This implies that if one considers three graphs $S' \subset S \subset \mathcal{G}$, the boundaries of S' in S and in $\mathcal{G}$ may differ. Recall that $\mathcal{N}(i)$ denotes the neighborhood of node i.

In the context of multi-agent systems, the nodes of the host graph $\mathcal{G}$ represent agents and the edges are communication links. In particular, an agent x has access to the states of all its neighbors and can use this piece of information to compute its control law. Although a complete graph is not necessary for a distributed control algorithm, we always assume that the host graph is connected.

In order to model the collective behavior of the agents we will use functions $f : V_{\mathcal{G}} \mapsto \mathbb{R}^d$ defined over a graph $\mathcal{G}$ [70]. The *partial derivative* of f is defined as $\partial_y f(x) \doteq f(y) - f(x)$ and enjoys the following properties:

(1) $\partial_y f(x) = -\partial_x f(y)$,

(2) $\partial_x f(x) = 0$,

(3) $\partial_y^2 f(x) = -\partial_y f(x)$.

The Laplacian of f is given by

$$\Delta f(x) \doteq -\sum_{y \in V_{\mathcal{G}}, y \sim x} \partial_y^2 f(x) = +\sum_{y \in V_{\mathcal{G}}, y \sim x} \partial_y f(x), \tag{84}$$

where the last identity follows from property (3). The integral and the average of f are defined, respectively, as

$$\int_G f \ dx \doteq \sum_{x \in V_{\mathcal{G}}} f(x), \quad \langle f \rangle \doteq \frac{1}{|V_{\mathcal{G}}|} \int_G f \ dx. \tag{85}$$

Let $L^2(\mathcal{G}|\mathbb{R}^d)$ be the Hilbert space composed by all functions $f : V_\mathcal{G} \mapsto \mathbb{R}^d$ endowed with the norm $\|f\|^2_{L^2} = \int_G \|f\|^2$. We will use the shorthand notation L^2 when there is no ambiguity on the underlying domain and range of the functions.

Let S be a subgraph of G and ∂S be its boundary in G. We assume that $S \cup \partial S = \mathcal{G}$. As in [70], we also consider the Hilbert space $H^1_0(S) = \{f \in L^2(\mathcal{G}) \ : \ f_{|\partial S} = 0\}$ (see [70] for the definition of a suitable norm on $H^1_0(S)$).

Note that a function $f \in H^1_0(S)$ is defined on $\bar{S}$ and possibly non null only on S.

The next theorem, proved in [70], characterize the eigenstructure of the Laplacian operator defined on $H^1_0(S)$.

Theorem 8.2.1. *Let $\mathcal{G}$ be a connected graph and S a proper subgraph of $\mathcal{G}$. Then, the operator $\Delta : H^1_0(S|\mathbb{R}^d) \mapsto L^2(\bar{S}|\mathbb{R}^d)$ has $|V_S|d$ strictly negative eigenvalues. Moreover, the corresponding eigenfunctions form a basis for $H^1_0(S|\mathbb{R}^d)$.*

8.3 Multiple Stationary Leaders

In this section we use PdEs for modeling and analyzing a group of agents with multiple leaders. A leader is just an agent that moves toward a predefined goal, and whose control policy is independent of the motion of all the followers. However, followers that are neighbors to the leader can use the leader state in order to compute their control inputs.

Let $r(x,t)$ be the position of the agent x at time $t \geq 0$, where[1] $r \in L^2$. The communication network is represented by the undirected and connected graph $\mathcal{G}$. For distinguishing between leaders and followers, we consider two subgraphs S_F and S_L of $\mathcal{G}$ such that $S_L = \partial S_F$ and $S_F \cup S_L = \mathcal{G}$, where the subscripts denote "Leaders" and "Followers" respectively. Note that we assume that all agents are either designated as leaders or followers.

As already mentioned in the introduction, we will assume that the followers obey

[1]For sake of conciseness, for a function $f(x,t) : V_\mathcal{G} \times \mathbb{R}^+ \to \mathbb{R}^d$ we will often write $f \in L^2$ instead of $f(\cdot,t) \in L^2$.

the simple dynamics $\dot{r}(x,t) = u(x,t)$, where

$$u(x,t) \doteq \Delta r(x,t) \tag{86}$$

is the *Laplacian* control law. Let $\hat{r}(x,t)$, $x \in V_{S_L}$ be the trajectory of the leaders. Then, the collective dynamics is represented by the model

$$\dot{r}(x,t) = \Delta r(x,t) \qquad x \in V_{S_F} \tag{87a}$$

$$r(x,t) = \hat{r}(x,t) \qquad x \in V_{S_L} \tag{87b}$$

endowed with the initial conditions $r(\cdot,0) = \tilde{r} \in L^2(S_F)$.

Model (87) is an example of a continuous-time Partial difference (PdE) with non-homogeneous Dirichlet boundary conditions. We refer the reader to [70–72] for an introduction to PdEs.

Being one of the most studied control paradigms for multi-agent systems, the Laplacian based feedback control allows the agents to achieve globally coordinated behaviors using only local information. The main results on Laplacian control available in the literature and specialized to model (87) are:

- in the leaderless case (i.e., $S_L = \emptyset$), the Laplacian control solves the rendezvous problem, i.e., $r(x,t) \to r^* \in \mathbb{R}^d$, $\forall x \in V_{\mathcal{G}}$ as $t \to +\infty$. Moreover, the agents converge exponentially to $r^* = \langle \tilde{r} \rangle$ thus achieving *average* consensus. These results have been established in [6, 73] through the joint use of tools in control theory and algebraic graph theory. A formal analysis of the PdE (87a) has been conducted in [71, 72, 74] showing a complete accordance with results available within the theory of the heat equation [75];

- in the case of a single leader (i.e., $V_{S_L} = \{x_L\}$) with fixed position (i.e., $\hat{r}(x_L,t) = \bar{r} \in \mathbb{R}^d$), Laplacian control solves the rendezvous problem with $r^* = \bar{r}$ [6]. This property has also been shown in [71, 72] within the PdE framework, thus

highlighting the profound links between model (87) and the heat equation with Dirichlet boundary conditions [75].

The first attempt of this paper is to characterize the asymptotic behavior of the followers in the presence of multiple leaders with fixed positions. To this end, for the remainder of this section, we will assume that $\hat{r}(x,t) = \bar{r}(x) \in L^2(S_L)$. The equilibria of (87) are then given by the solutions to the PdE

$$\Delta h(x) = 0 \qquad x \in V_{S_F} \tag{88a}$$

$$h(x) = \bar{r}(x) \quad x \in V_{S_L} \tag{88b}$$

and they have been studied in [70]. In particular, [70, Theorem 3.5] shows that if the hosting graph $\mathcal{G}$ is connected and $V_{S_L} \neq \emptyset$ then, the PdE (88) has a unique solution[2] $h(x)$. By analogy with the jargon of Partial Differential Equations, h is termed the *harmonic extension of the boundary conditions* $\bar{r}$.

Our next aim is to verify that $r \to h$ as $t \to +\infty$. Let us consider the decomposition

$$r(x,t) = r_0(x,t) + h(x), \quad r_0 \in H_0^1(S_F) \tag{89}$$

Since h does not depend upon time and $\Delta h = 0, \forall x \in V_{S_F}$, the PdE (87) is equivalent to the following one

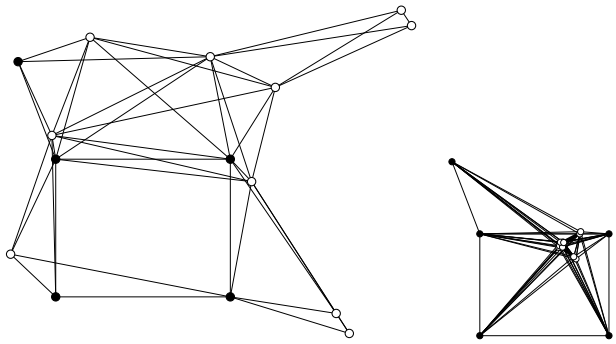
$$\dot{r}_0(x,t) = \Delta r_0(x,t) \qquad x \in V_{S_F} \tag{90a}$$

$$r_0(x,t) = 0 \qquad x \in V_{S_L} \tag{90b}$$

From (89), it is apparent that the problem of checking if $r \to h$ as $t \to +\infty$ can be recast into the problem of studying the convergence to zero of the solutions to the PdE (90). The fact that $r_0 \to 0$ as $t \to +\infty$ follows from Theorem 8.2.1 and it can be shown by proceeding exactly as in the proof of [74, Theorem 5][3].

[2] [70, Theorem 3.5] assumes that the subgraph S is *induced* (see [70] for the definition of induced subgraphs). However, a careful examination of the proof, reveals that this assumption is unnecessary.

[3] Actually, [74, Theorem 5] proves a stronger property, namely that the origin of (90) is "exponentially stable on the space $H_0^1(S)$". The definition of stability of equilibria on subspaces is provided in [74].



(a) (b)

Figure 26. An example of the application of Theorem 8.3.1 is given. Initially, some of the followers (white) are located outside Ω_L but after a while they have all reached Ω_L, spanned by the stationary leaders (black). The edges between agents capture the information flow in this static interaction graph.

The next Theorem, proved in Appendix A, highlights a key geometrical feature of $h(x)$. For a set X of points in $\mathbb{R}^d$, $\mathrm{Co}(X)$ will denote its convex hull. Moreover, the set Ω_L is the convex hull of leaders positions, i.e., $\Omega_L \doteq \mathrm{Co}(\{\bar{r}(y),\ y \in V_{S_L}\})$.

Theorem 8.3.1. *Let S_1 be a nonempty connected subgraph of S_F and ∂S_1 be its boundary in $\mathcal{G}$. Then, $\forall x \in V_{S_1}$ it holds*

$$h(x) \in \mathrm{Co}(\{h(y),\ y \in V_{\partial S_1}\}). \tag{91}$$

Moreover, one has that $h(x) \in \Omega_L$, i.e., that the position of each follower lies in the convex hull of the leaders positions. Finally, if Ω_L is full dimensional, then $h(x) \in \Omega_L \backslash \partial\Omega_L$, $\forall x \in V_{S_F}$.

This result is illustrated in Figure 26. We start by introducing a basic result on polytopes.

Lemma 8.3.2. *Consider the polytope $P = \mathrm{Co}(X)$ where $X = \{x_i \in \mathbb{R}^d : i = 1, \ldots, L\}$ and let X_1 be a proper subset of X. If $x \in \mathrm{Co}(X_1)$, $\forall x \in X \backslash X_1$, then $P = \mathrm{Co}(X_1)$.*

Proof. The conditions $x \in X \backslash X_1$ and $x \in \mathrm{Co}(X_1)$ imply that x is not a vertex of P. Then, X_1 includes all vertexes of P, thus proving that $P = \mathrm{Co}(X_1)$. ∎

Lemma 8.3.3. *Let G be a host graph, S a subgraph of G and T_1 a proper subgraph of S and ∂T_1 the boundary of T_1 in G. Consider $\bar{x} \in V_{\partial T_1} \cap V_S$, and let $r \in L^2(G)$ be a function verifying*

$$r(\bar{x}) \in \mathrm{Co}(\{r(y) : y \in \mathcal{N}(\bar{x})\}) \tag{92}$$

$$r(x) \in \mathrm{Co}(\{r(y) : y \in V_{\partial T_1}\}), \quad \forall x \in V_{T_1} \tag{93}$$

Let T_2 be the subgraph associated with $V_{T_1} \cup \{\bar{x}\}$ and ∂T_2 be the boundary of T_2 in G. Then, for all $x' \in V_{T_2}$ it holds

$$r(x') \in \mathrm{Co}(\{r(y) : y \in V_{\partial T_2}\}). \tag{94}$$

Proof. From (92), one has that all $\bar{r} \in \{r(y) : y \in V_{T_2}\}$ verify $\bar{r} \in P$ where $P = \mathrm{Co}(\{r(y) : y \in V_{\partial T_1} \cup \mathcal{N}(\bar{x})\})$. In particular, if $x \in \mathcal{N}(\bar{x}) \cap T_1$ one has that $r(x) \in \mathrm{Co}(\{r(y) : y \in V_{\partial T_1}\})$. Recalling (93) and that $\bar{x} \in V_{\partial T_1}$ one can apply Lemma 8.3.2 and obtain

$$P = \mathrm{Co}(\{r(y) : y \in (V_{\partial T_1} \backslash \{\bar{x}\}) \cup (\mathcal{N}(\bar{x}) \backslash V_{T_1})\}).$$

The proof is concluded by realizing that

$$V_{\partial T_2} = (V_{\partial T_1} \backslash \{\bar{x}\}) \cup (\mathcal{N}(\bar{x}) \backslash V_{T_1}).$$

■

We are now in a position to prove Theorem 8.3.1.

Proof. (Theorem 8.3.1.)

Let $p = x_0 x_1 \dots x_L$ be a path going through all nodes of S_1. Since $\Delta h(x) = 0$, $\forall x \in V_{S_F}$ from (84) one has

$$h(x) = \frac{1}{|\mathcal{N}(x)|} \sum_{y \in \mathcal{N}(x)} h(y), \ \forall x \in V_{S_F}$$

which implies that

$$h(x) \in \mathrm{Co}\{h(y) : y \in \mathcal{N}(x)\} \tag{95}$$

We will prove the theorem using a recursive argument on the nodes composing $\mathcal{N}$. First, note that $x_1 \in \mathcal{N}(x_0)$. Let T_1 and T_2 be the subgraphs associated with $\{x_0\}$ and $V_{S_1} \cup \{x_1\}$, respectively. Lemma (8.3.3) can be applied with $\bar{x} = x_1$. Indeed, (92) amounts to (95) for $x = x_0$ and (93) amounts to (95) for $x = x_1$. Then, from (94) we have

$$h(x) \in \mathrm{Co}(\{h(y) : y \in V_{\partial T_2}\}),\ \forall x \in V_{T_2}$$

Now, we denote by $S^{(i)}$, $i < L$ the subgraph of S_1 associated with the $i+1$ nodes $\{x_0, x_1, \ldots, x_i\}$ and by $\partial S^{(i)}$ its boundary in G. Assume now that at the i-th step, $i < L$ we have

$$h(x) \in \mathrm{Co}(\{h(y) : y \in V_{\partial S^{(i)}}\}),\ \forall x \in V_{S^{(i)}} \tag{96}$$

We need to prove that:

$$h(x) \in \mathrm{Co}(\{h(y) : y \in V_{\partial S^{(i+1)}}\}),\ \forall x \in V_{S^{(i+1)}}. \tag{97}$$

Note that $x_{i+1} \in \mathcal{N}(x_i)$. Set $T_1 = S^{(i)}$ and let T_2 be the graph associated with $V_{S^{(i)}} \cup \{x_{i+1}\}$. Lemma (8.3.3) can be applied with $\bar{x} = x_{i+1}$. Indeed, (92) amounts to (95) for $\bar{x} = x_{i+1}$ and (93) amounts to (96). Then, from (94) we have

$$h(x) \in \mathrm{Co}(\{h(y) : y \in V_{\partial T_2}\}),\ \forall x \in V_{T_2}.$$

Since, $T_2 = S^{(i+1)}$, formula (91) is proved. If S is connected, the result holds also for $S = S_1$. If S is not connected, we apply (91) on each connected component S_i, $1 \leq i \leq n$, and, by simple algebra, obtain

$$h(x) \in \mathrm{Co}(\{h(y),\ y \in V_{\partial S_1} \cup V_{\partial S_2} \cup \ldots \cup V_{\partial S_n}\}).$$

The proof that each follower lies in the convex hull of the leaders positions is ended by realizing that $V_{\partial S} = V_{\partial S_1} \cup V_{\partial S_2} \cup \ldots \cup V_{\partial S_n}$.

The fact that the full dimensionality of Ω_L implies that $h(x) \in \Omega_L \backslash \partial\Omega_L$, $\forall x \in V_{S_F}$ is proved by contradiction. Let $x \in V_{S_F}$ be such that $h(x) \in \partial\Omega_L$ and denote

with χ the supporting hyperplane of Ω_L such that $h(x) \in \Omega_L \cap \chi$. Then, since $h(x) \in \mathrm{Co}(\{h(y),\ y \in \mathcal{N}(x)\})$, all $y \in \mathcal{N}(x)$ verify $h(y) \in \partial\Omega_L \cap \chi$. Iterating the argument over the followers lying on χ, one would find that also all leaders $x \in V_{S_L}$ lie on χ and this contradicts the fact that Ω_L is full dimensional. ■

Another geometrical feature which we need is the following:

Theorem 8.3.4. *Suppose that Ω_L is fully dimensional and that $r(x,t)$ is evolving according to (87). Suppose that, at a given time $t = \overline{t}$, there is an agent $x \in V_{S_F}$ such that $r(x,\overline{t}) \in \partial\Omega_L$. Then, two situations may occur:*

1. *there exists an (affine) hyperplane χ such that*

$$r(x,\overline{t}) \in \chi \cap \partial\Omega_L, \text{ and } r(y,\overline{t}) \in \chi \cap \partial\Omega_L \ \forall y \in \mathcal{N}(x).$$

Then:

$$\exists\, \alpha > 0 \ : \ r(x,\overline{t}) + \alpha \dot{r}(x,\overline{t}) \in \chi \cap \partial\Omega_L, \tag{98}$$

2. *otherwise,*

$$\exists\, \alpha > 0 \ : \ r(x,\overline{t}) + \alpha \dot{r}(x,\overline{t}) \in \Omega_L \setminus \partial\Omega_L. \tag{99}$$

Note that (98) means that the velocity of x will be along the hyperplane χ (in other words, the agent may slide on the boundary $\partial\Omega_L$), whereas (99) means that the velocity of x is pointing inside the polytope Ω_L.

Proof. (Theorem 8.3.4)

Since $r(x,t)$ obeys to (87), by rearranging terms we obtain:

$$\dot{r}(x,\overline{t}) = -|\mathcal{N}(x)| r(x,\overline{t}) + \sum_{y \in \mathcal{N}(x)} r(y,\overline{t}).$$

Then, setting $\alpha = |\mathcal{N}(x)|^{-1}$, it holds:

$$r(x,\overline{t}) + \alpha \dot{r}(x,\overline{t}) = |\mathcal{N}(x)|^{-1} \sum_{y \in \mathcal{N}(x)} r(y,\overline{t}),$$

i.e., $r(x,\overline{t}) + \alpha\dot{r}(x,\overline{t})$ is the barycenter $b(\mathcal{Y}_x)$ of the polytope $\mathcal{Y}_x \doteq \text{Co}(\{r(y,\overline{t}),\ y \in \mathcal{N}(x)\})$. Note that: first $\mathcal{Y}_x \in \Omega_L$, second, thanks to convexity, the barycenter of $\mathcal{Y}_x$ lies in the relative interior of $\mathcal{Y}_x$. Thus, if all $y \in \mathcal{N}(x)$ verify that $r(y,\overline{t}) \in \chi \cap \partial\Omega_L$ then $\mathcal{Y}_x \subset \chi \cap \partial\Omega_L$ and so does $b(\mathcal{Y}_x)$, i.e., $b(\mathcal{Y}_x) \in \chi \cap \partial\Omega_L$; otherwise $b(\mathcal{Y}_x) \in \Omega_L \backslash \partial\Omega_L$.

■

8.4 Leader-Follower Containment Control

Containment of all the followers is achieved in the case of static leaders in last section. However, if the leaders are moving, this property might be violated. In order to prevent the followers from leaving the polytope spanned by the leaders, appropriate control strategy need to be designed for the leaders to guarantee the containment. In what follows, we propose a hybrid strategy for this purpose and analyze the liveness and reachability.

8.4.1 Hybrid Control Strategy

For the sake of containment, we define two distinctly different control modes for the evolution of the leaders. The first of the two control modes is the $STOP$ mode. As the name indicates, this mode corresponds to the leaders halting their movements altogether in order to prohibit a break in the containment:

$$
\begin{aligned}
&STOP: \\
&\dot{r}(x,t) = \Delta r(x,t) \qquad x \in V_{S_F} && \text{(100a)}\\
&r(x,t) = \hat{r}(x,t) \qquad x \in V_{S_L} && \text{(100b)}\\
&\dot{\hat{r}}(x,t) = 0 \qquad x \in V_{S_L} && \text{(100c)}
\end{aligned}
$$

It is clear that in order to execute this mode, no information is needed for the leaders whatsoever.

The second control mode under consideration is the GO mode, in which the leaders move toward a given target formation. A number of different control laws can be

defined for this, but, for the sake of conceptual unification, we let the GO mode be given by a Laplacian-based control strategy as well.

$$
\begin{aligned}
&GO: \\
&\dot{r}(x,t) = \Delta r(x,t) && x \in V_{S_F} && \text{(101a)}\\
&r(x,t) = \hat{r}(x,t) && x \in V_{S_L} && \text{(101b)}\\
&\dot{\hat{r}}(x,t) = \Delta_{S_L}(\hat{r}(x,t) - r_T(x)) && x \in V_{S_L} && \text{(101c)}
\end{aligned}
$$

where $r_T(x),\ x \in V_{S_L}$ denotes the desired target position of leader x, and where we use Δ_{S_L} to denote the Laplacian operator defined solely over the subgraph S_L, i.e.,

$$\Delta_{S_L} f(x) \doteq - \sum_{y \sim x,\ y \in V_{S_L}} \partial_y^2 f(x).$$

Under the assumption that S_L is connected, and by exactly the same reasoning as for the standard rendezvous problem, under the influence of the GO mode alone the leaders will converge exponentially to $r_L(x) = \langle \hat{r}(\cdot,0) \rangle - r_T(x)$, i.e., $\exists k > 0, \eta > 0$ such that $\|\hat{r}(\cdot,t) - r_L(x)\|_{L^2} \leq k e^{-\eta t} \|\hat{r}(\cdot,0) - r_L(x)\|_{L^2}$. In other words, no convergence to a predefined point is achieved. Rather, this control law ensures that the leaders arrive at a translationally invariant target formation.

Note that the details of the leaders' motion is not crucial and this particular choice is but one of many possibilities. However, this choice is appealing in that it makes the information flow explicit, and the leaders only need access to the positions (and target locations) of their neighboring leaders in order to compute their motion. As such the decentralized character of the algorithm is maintained.

In order for fully specify the hybrid Stop-Go leader policy depicted in Figure 27, transition rules are needed as well. As before, let Ω_L denote the leader-polytope and let $d(\mu, \Omega_L)$ denote the signed distance

$$d(\mu, \Omega_L) \doteq \zeta_{\Omega_L}(\mu) \min_{x \in \partial\Omega_L} \|\mu - x\|_2, \tag{102}$$

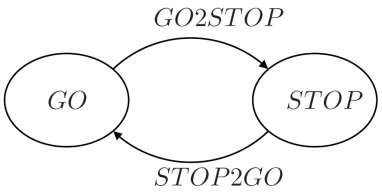


Figure 27. The hybrid automaton implementing the Stop-Go policy.

where $\|\cdot\|_2$ denotes the Euclidean 2-norm, and where $\zeta_{\Omega_L}(\mu) = -1$ if $\mu \in \Omega_L$ and $+1$ otherwise. Using this distance measure we let the two guards be given by

$$GO2STOP : \exists y \in V_{S_F} \mid d(r(y,t), \Omega_L) \geq 0? \tag{103a}$$

$$STOP2GO : d(r(y,t), \Omega_L) < -\epsilon \ \forall y \in V_{S_F}? \tag{103b}$$

where $\epsilon > 0$ is a threshold.

Note that the guard $STOP2GO$ is crossed only if the following assumptions are verified:

Assumption 8.4.1. Let $\hat{h}(\cdot, t)$ be the solution to (88) for $\bar{r}(\cdot) = \hat{r}(\cdot, t)$, $\forall t \geq 0$ and consider the set $\Omega_L^\epsilon(t) = \{y \in \Omega_L(t) : \text{dist}(y, \partial\Omega_L(t)) > \epsilon\}$. Then

1. $\Omega_L^\epsilon(t)$ is nonempty, $\forall t \geq 0$;

2. $\text{Co}(\{\hat{h}(x,t), x \in V_{S_F}\}) \subset \Omega_L^\epsilon(t)$.

In particular, Assumption 8.4.1 implies that Ω_L must be full-dimensional at all times and "sufficiently fat" along every direction. Conditions relating property 2 of Assumption 8.4.1 to the graph topology are currently under investigation. A few comments must be made about the computation and communication requirements that these guards give rise to. If two leaders are located at the end-points of the same face of Ω_L, then they must be able to determine if any of the followers are in fact on this face. This can be achieved through a number of range sensing devices, such as ultrasonic, infra-red, or laser-based range-sensors. Moreover, in order for all leaders to transition between modes in unison, they must communicate between them, which means that either S_L is a complete graph, or that multi-hop strategies are needed.

In either way, a minimal requirement for these mode transitions to be able to occur synchronously, without having to rely on information flow across follower-agents, is that S_L must be connected.

The hysteresis threshold $\epsilon > 0$ in the $STOP2GO$ guard (see Figure 28) and the next assumption are needed in order to avoid Zeno behaviors. Let ρ_{Ω_L} denote the supreme of the diameter of Ω_L during an execution.

Assumption 8.4.2. $\exists M < \infty$ such that $\rho_{\Omega_L} \leq M$.

It is easy to check that Assumption (8.4.2) is verified when Laplacian control governs the leaders' motion in the GO mode as in (101c). Indeed, the exponential convergence of $\hat{r}(x,t)$ to $r_L(x) = \langle \hat{r}(\cdot,0) \rangle - r_T(x)$ implies that $\hat{r}(x,t)$ is bounded at all times. However, Laplacian control is but one of many possible control strategies and can be replaced by other control schemes (e.g. plan-based leader control laws) without generating Zeno executions as long as Assumption 8.4.2 is verified.

Theorem 8.4.1. *Under Assumptions 8.4.2 and 8.4.1, the hybrid automaton in Figure 27 defined by* (100)*,* (101) *and* (103) *is non-Zeno.*

Proof. Let the system be in the $STOP$ mode. Under Assumption 8.4.2 we have

$$\begin{aligned} \|\dot{r}(x,t)\| = \|\Delta r(x,t)\| \leq \sum_{y \sim x} \|\partial_y r(x)\| \leq \sum_{y \sim x} \rho_{\Omega_L} \leq N \rho_{\Omega_L}, \\ \forall x \in V_{S_F}. \end{aligned} \tag{104}$$

From Assumption 8.4.1, in order for the system to leave the $STOP$ mode, at least one follower agent must have traveled at least a distance ϵ, which in turn implies that the system will always stay for a time greater than or equal to $\epsilon / N \rho_{\Omega_L}$ in the $STOP$ mode. In order for the system to exhibit Zeno executions, a necessary condition is that the difference between the transition times must approach zero [76]. Since this is not the case here, the non-Zeno property is established. ∎

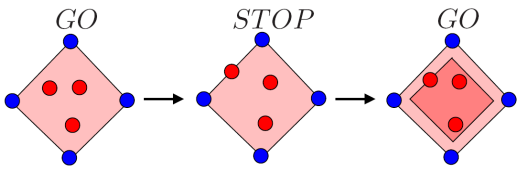


Figure 28. A hysteresis-based transition strategy avoids Zeno executions.

8.4.2 An Example

An execution of the hybrid automaton is shown in Figure 52, where three leaders (black) maneuver four followers (white). The initial position and final position of the leaders are $r(x,0) = \{(1,-3),(0,-1),(0,1)\}$ and $r_T(x) = \{(0,-2),(1,2),(2,-2)\}$ respectively.

During the maneuvering, the Stop-Go policy defined by the guards (103) is adopted.

The magnitude of the velocities of the agents are shown in Figure 30, where we can see the instances when the leaders stop to make sure that the followers remain inside the leader-polytope. The snap-shots of the transition instances are shown in Figure 31.

8.4.3 Liveness and Reachability

As already mentioned, the proposed solution is non-Zeno. However, as it is currently defined, the Stop-Go policy may be blocking in the sense that the system never leaves the $STOP$ mode. One remedy to this problem is to allow the containment to be slightly less tight. In other words, we can select different guards, e.g.

$$GO2STOP : \exists y \in V_{S_F} \mid d(r(t,y),\Omega_L) > 2\delta? \tag{105a}$$

$$STOP2GO : d(r(t,y),\Omega_L) \leq \delta \ \forall y \in V_{S_F}? \tag{105b}$$

where $\delta > 0$. What this means is that we do not enter the $STOP$ mode until a follower is 2δ outside Ω_L. Let us define

$$\Omega_{L,\delta} \doteq \{y \in \mathbb{R}^d : \text{dist}(y,\Omega_L) \leq \delta\}$$

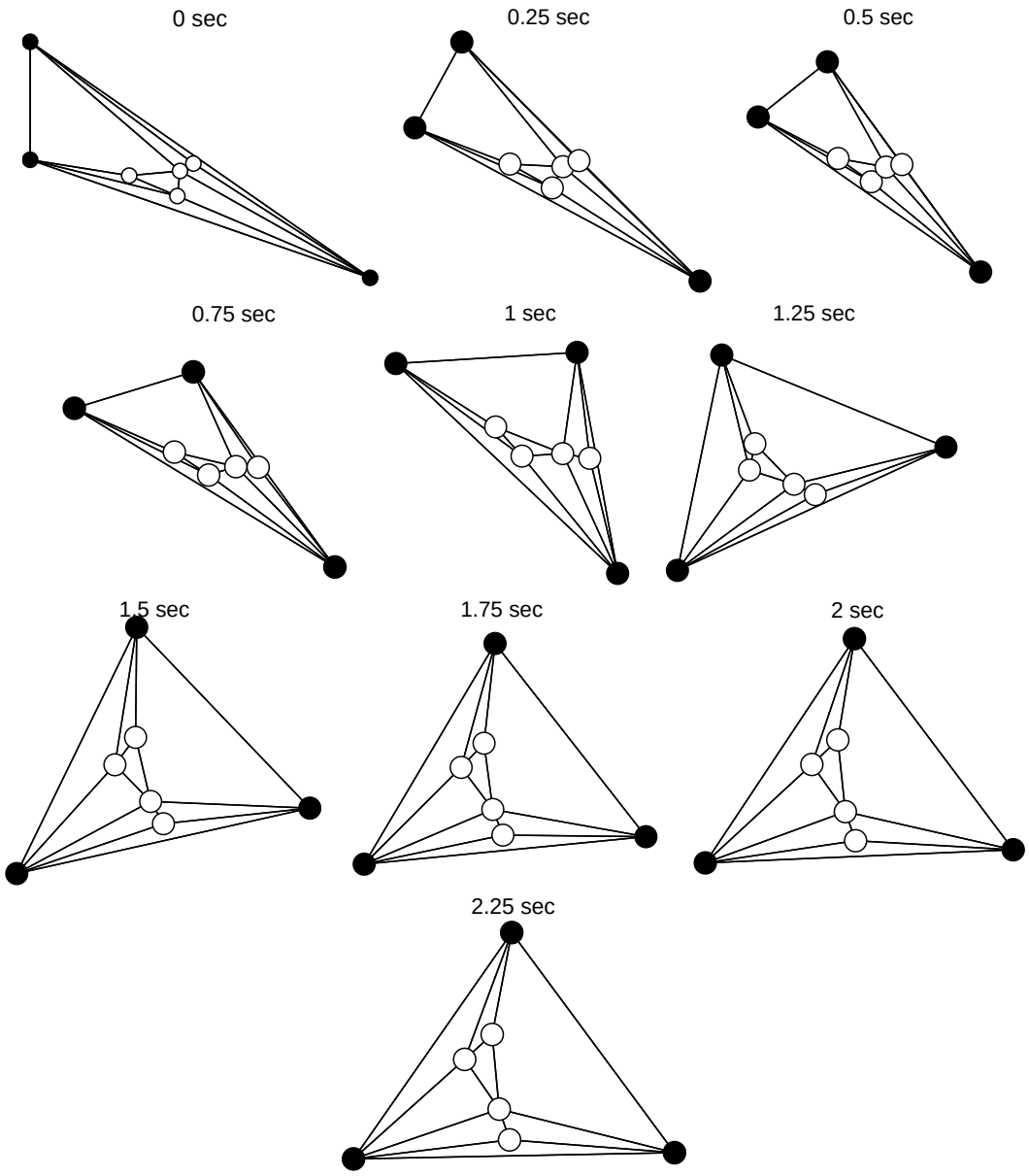


Figure 29. A containment process where 4 followers (white) are guided by 3 leaders (black), who use the hybrid Stop-Go control policy.

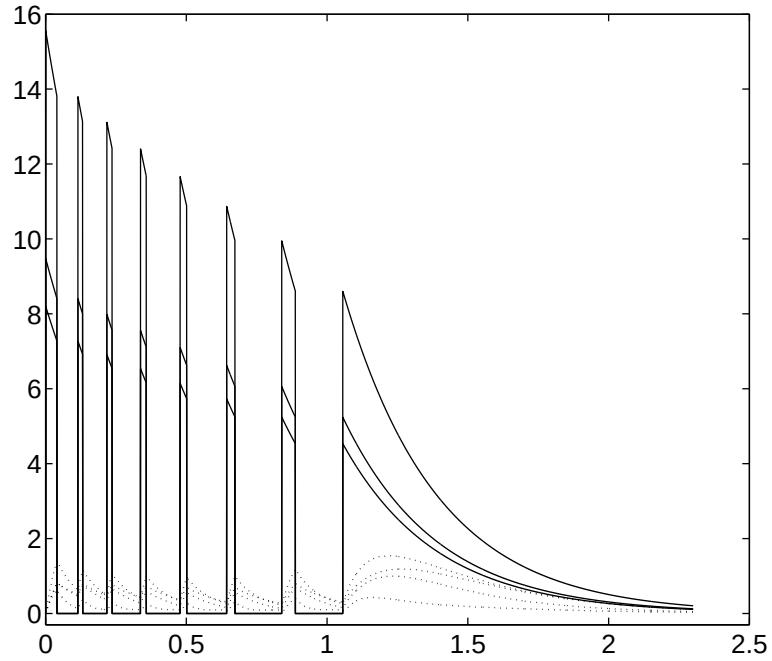


Figure 30. Agents' speeds vs. time (sec) in the containment process. Solid lines correspond speeds of the leaders while dashed lines correspond to those of the followers.

Note that, from (102), one has $\Omega_L \subset \Omega_{L,\delta}$. The next Theorem summarizes the main properties of the resulting hybrid automaton. A remarkable feature of the guards (105) is that Assumption 8.4.1 is no longer needed in order to guarantee liveness.

Theorem 8.4.2. *Under Assumption 8.4.2, the hybrid automaton in Figure 27 defined by* (100), (101) *and* (105) *is non-Zeno, live, in the sense of always leaving the STOP mode eventually, and convergent in the sense that* $\hat{r}(x,t) \to \langle \hat{r}(\cdot,0) \rangle - r_T(x)$.

Proof. We first prove liveness. Assume that the system is in the $STOP$ mode. From Theorem 91, we have that $h \in \Omega_L$. Since $\forall x \in S_F$, $r(x,t) \to h$, and $\Omega_L \subset \Omega_{L,\delta}$, every follower will eventually get back in $\Omega_{L,\delta}$ in finite time (recall that the leaders are stationary in the $STOP$ mode) hence triggering a transition to the GO mode.

Under Assumption 8.4.2, it holds $\|\dot{r}(x,t)\| \leq N(\rho_{\Omega_L} + 2\delta)$ and we can repeat the non-Zeno argument in the proof of Theorem 8.4.1 in order to see that the system always stays in the GO mode for a time greater than or equal to $\delta/(N(\rho_{\Omega_L} + 2\delta))$.

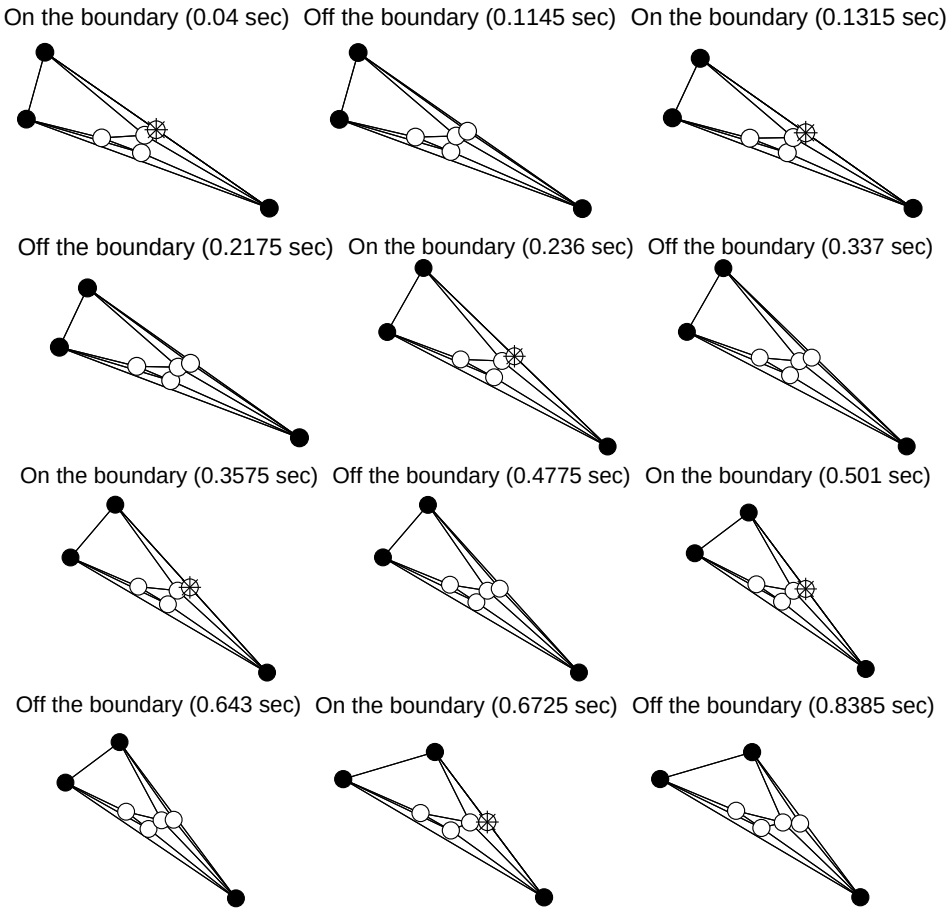


Figure 31. Time instances when transitions between the GO and $STOP$ mode occur. (The asterisk denotes the particular follower who intersects the boundary.)

As a result, in a non-blocking system the leaders will be given infinitely many opportunities to move during a finite (bounded away from zero) time horizon, which implies convergence to the target location as long as the leaders would in fact end up at the target location under the influence of the *GO* mode alone. ■

8.5 Hierarchial Containment Control

Since the two-layer structure previously proposed can be extended to multiple layers quite naturally, we investigate this issue in the next sections. In other words, we will impose a hierarchical structure on the network topology, which can be viewed as a cascade of the leader-follower structure presented before.

8.5.1 Hierarchical Network Topologies

In the hierarchial structure, the agents are organized into M layers encoded through subgraphs. The inner-most layer is layer 1 and the outermost layer is layer M. We moreover assume that the network topology is such that

$$\bigcup_{i=1}^{M} V_{S_i} = V_{\mathcal{G}},\ V_{S_i} \cap V_{S_j} = \emptyset,\ i \neq j$$

and

$$V_{\partial S_{i-1}} \subseteq V_{S_i} \cup V_{S_{i-2}},\ \forall\ i = 3, \cdots, M$$
$$V_{\partial S_1} \subseteq V_{S_2}$$
$$V_{\partial S_M} \subseteq V_{S_{M-1}},$$

where S_i denotes the subgraph corresponding to layer i and ∂S_{i-1} is the (non-empty) boundary of S_{i-1} in the host graph $\mathcal{G}$. Moreover, we assume that each subgraph is connected, and an example of such a topology is given in Figure 32.

The main reason why such a layered approach is beneficial comes from the amount of information the different agents have to store. To see this, consider a generalization of the hybrid Stop-Go policy to affect agents in all layers except those at the inner-most layer (layer 1). Let also Ω_i be the polytope spanned by the positions of agents

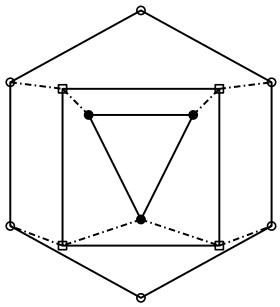


Figure 32. A hierarchical layering of the network topology into three layers. The solid lines correspond to intra-layer edges while the dash-dotted lines correspond to inter-layer edges.

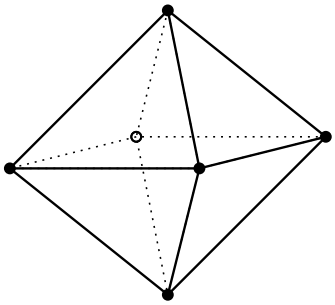


Figure 33. Layer i in 3D, where each agent forms a face in Ω_i with all but one agent.

in layer i. In order to execute such a hybrid control policy, the individual agents in layer i must keep track of the agents in layer $i-1$ as well as its neighbors in layer i with whom they form faces in Ω_i.

In what follows, we assume, for notational simplicity, that we have a total of N agents, equally distributed across M layers. In other words, we have a total number of N/M agents in each layer. If the agents are planar, i.e., they move in a 2D space, each agent in V_{S_i} can at most form faces in Ω_i with two other agents in V_{S_i}. However, each agent also has to keep track of all the agents in layer $i-1$, which implies that the total number of agents that each agent must keep track of (not considering agents in the inner-most layer) is

$$\mu_{2,M,N} = N/M + 3,$$

where the subscript 2 denotes the planar case. The first term corresponds to the agents in layer $i-1$, while the 3 corresponds to the two neighbors in Ω_i as well as the agent's

own position. This number should be complemented with a situation where we only use two layers, and since $\mu_{2,M,N}$ is monotonously decreasing in M, we have that the total amount of information that must be stored by each agent (not in the inner-most layer) is strictly less in the hierarchical situation since $\mu_{2,M,N} < \mu_{2,2,N},\ \forall M > 2$.

Now, if the agents are evolving in $\mathbb{R}^n$, where $n > 2$, each agent may be forced to keep track of the positions of all but one of the agents in its layer, as shown in Figure 33. In other words, if we let $\bar{\mu}$ denote this worst-case scenario, we get the similar result

$$\bar{\mu}_{n,M,N} = 2N/M - 1 < \bar{\mu}_{n,2,N} = N - 1,\ \forall M > 2.$$

The conclusion to draw from this is that for each individual agent in the outer-most layer, a hierarchical structure is to prefer. However, this structure implies that agents not located in the outer-most layer will have to be more advanced as well in that they are executing a more complex control policy. This is both due to the fact that they need to store information about other agents, but they also must execute a more advanced strategy compared to the non-hierarchical case. As such, a hierarchical topology reduces the demands on the most advanced agents while it increases the demands on the less advanced agents, thus acting as an equalizer of the capabilities needed by the different agents.

8.5.2 Multi-Layered Containment Control

Recall that the control strategy presented in Section 8.4 is given by a hybrid Stop-Go control policy for the leaders, while the followers execute a simple, distributed control law. The basic idea is to ensure that the leaders come to a halt whenever a follower leaves the convex polytope Ω_L spanned by the leader, giving the followers the chance to return to Ω_L. In the multi-layer scenario, this containment strategy will be generalized through the convex polytopes

$$\Omega_{i,\alpha} \doteq \{y \in \mathbb{R}^d : \text{dist}(y, \Omega_i) \leq \alpha\}, i = 2, \ldots, M.$$

First, the movements of the agents in the inner-most layer are dictated by the closed-loop dynamics

$$\dot{r}(x,t) = \Delta_1 r(x,t) \quad x \in V_{S_1}. \tag{106}$$

where the reduced Laplacian $\Delta_k,\ k = 1, \ldots, M-1$ is defined as

$$\Delta_k r(x,t) \doteq - \sum_{y \in V_{S_k} \cup V_{S_{k+1}}, y \sim x} \partial_y^2 r(x,t). \tag{107}$$

Now, since $V_{\partial S_1} \subseteq V_{S_2}$, the only agents outside V_{S_1} that agents in V_{S_1} are connected to are those in V_{S_2}. As we have shown in Section 8.4, if the agents in V_{S_2} are fixed and stationary, and $S_1 \cup S_2$ is connected, the agents in V_{S_1} will converge to Ω_2, i.e., to the convex polytope spanned by the agents in layer 2, under the reduced Laplacian control law (106).

Next, agents in layers $2, \ldots, M-1$ will execute a Laplacian Stop-Go policy in that agents in layer i will switch between executing a Laplacian control law when all agents in layer $i-1$ are in $\Omega_{i,\delta}$ (GO_i mode) to halting their evolution if an agent in layer $i-1$ leaves $\Omega_{i,2\delta}$ ($STOP_i$ mode). As show in Section 8.4.3 for the two-layer case, these guards are chosen in order to avoid blocked executions. The control strategy for layer $i, i = 2, \ldots, M-1$, can be summarized as

$$\dot{r}(x,t) = \begin{cases} \Delta_i r(x,t), & GO_i; \\ 0, & STOP_i. \end{cases} \qquad x \in V_{S_i}. \tag{108}$$

The only remaining agents are the leaders, i.e., the agents in layer M. They will also halt their execution ($STOP_M$ mode) if agents in layer $M-1$ are outside $\Omega_{M,2\delta}$.

$$STOP_M: \quad \dot{r}(x,t) = 0, \quad x \in V_{S_M} \tag{109}$$

The only difference is that these agents will be given a goal-oriented motion in the GO_M mode. Similarly to (101c), we let the GO_M mode be given by a Laplacian-based control strategy as well:

$$GO_M: \quad \dot{r}(x,t) = \Delta_M(r(x,t) - r_T(x)), \quad x \in V_{S_M}, \tag{110}$$

where $r_T(x),\ x \in V_{S_M}$ denotes the desired target position of leader x, and the "Leader Laplacian" Δ_M is defined as

$$\Delta_M r(x,t) \doteq - \sum_{y \sim x,\ y \in V_M} \partial_y^2 r(x,t).$$

As for the PdE (101c), if S_M is connected, the leaders will converge exponentially to $-r_T(x) + \frac{1}{|V_{S_M}|} \int_{S_M} r(x,0)$ under the effect of the GO mode alone.

8.5.3 An Example

As an example for multi-layered containment control, the evolution of a 9-agent 3-layer structure is shown in Figure 34. The outer layer is asked to rotate by $\frac{2}{3}\pi$ radians. The switching sequences of both layers are shown in Figure 35.

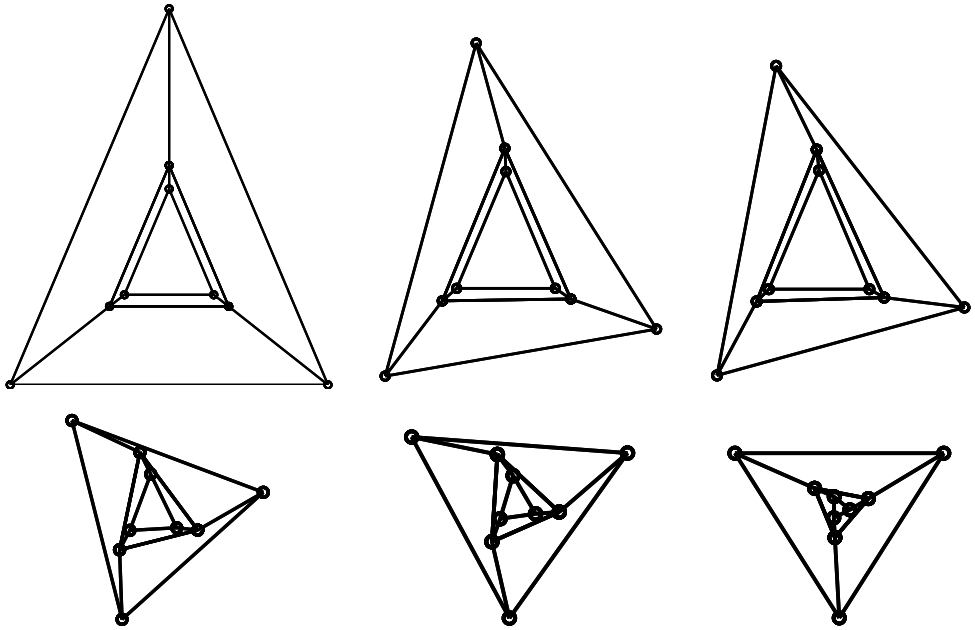


Figure 34. Hierarchical, multi-layered containment control with 3 layers and 3 agents in each layer. Here we choose the threshold $\delta = 0.01$

8.6 Complexity and Performance in Choosing the Optimal Layer Size

In this section, we will discuss how the multi-layered structure affects the performance of the multi-agent group. Two major issues will be considered, namely structural

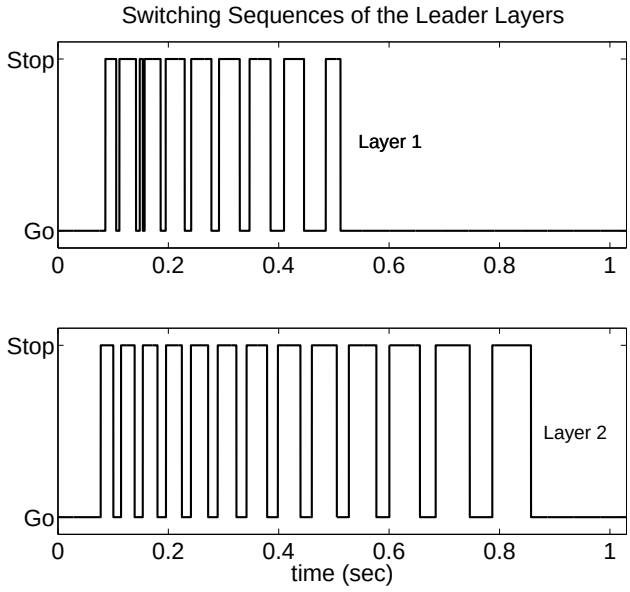


Figure 35. The switching sequences associated with Layers 1 and 2 during the process depicted in Figure 34.

complexity and convergence speed.

Structural complexity measures of a network topology capture certain aspects of the information flow through the network. For instance, the *Kirchhoff complexity*, $C_K(\mathcal{G})$, of a graph $\mathcal{G}$ is given by

$$C_K(\mathcal{G}) = \log(\tau(\mathcal{G})),$$

where $\tau(\mathcal{G})$ is the number of spanning trees in $\mathcal{G}$. And, by Kirchhoff's theorem [77], $\tau(\mathcal{G})$ is related to the spectrum of the so-called graph Laplacian through

$$\tau(\mathcal{G}) = \frac{1}{N}\prod_{i=2}^{N}\lambda_i(\mathcal{G}).$$

Here $\lambda_i(\mathcal{G})$ is the ith smallest eigenvalue of the *graph Laplacian* $\mathcal{L}(\mathcal{G}) = \mathcal{D} - \mathcal{A} \in \mathbb{R}^{N\times N}$ where $\mathcal{D} = \text{diag}\{d_1, \ldots, d_N\}$, d_i is the degree of the i-th node of $\mathcal{G}$ and $\mathcal{A}$ is the adjacency matrix of the graph $\mathcal{G}$ [78][4].

In addition to the Kirchhoff complexity, we will also study a complexity measure

[4]The relationship between graph Laplacian an the operator Δ has been discussed in [72].

based on the shortest pairwise path length between the agents [27], which is given by

$$C_A(\mathcal{G}) = \sum_{x \in V(\mathcal{G})} (d_x + \sum_{y \in V(\mathcal{G}) \setminus \mathcal{N}(x)} \frac{d_y}{k_{xy}}),$$

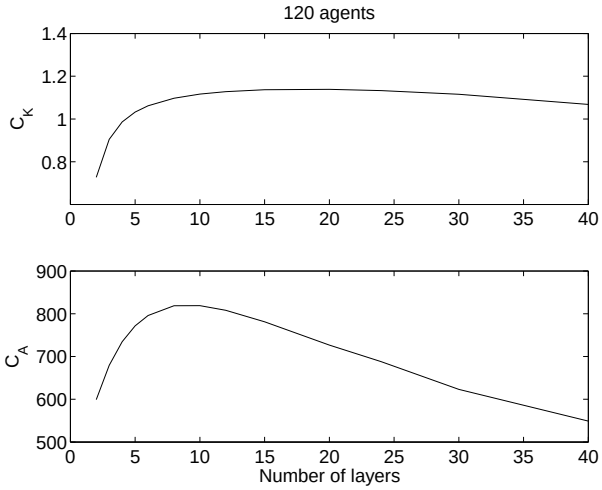
where d_x is the degree of node x and k_{xy} is the length of the shortest path between x and y, and recall that $\mathcal{N}(x)$ denotes the set of neighbors to x.

Figure 36 shows examples where agents are equally distributed among multiple layers. In fact, we will only study hierarchical networks of this type in the remainder of the paper. The reason for this is that they allow a straightforward comparison between the number of agents and layers. However, it should be pointed out that the proposed complexity and performance measures are applicable to arbitrary networks.

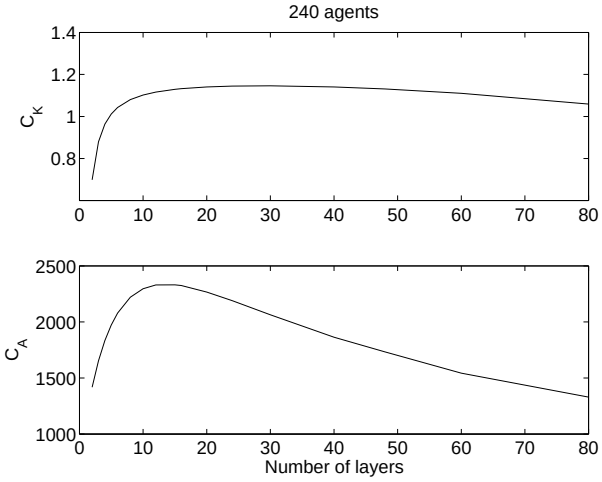
In Figure 36, the Kirchhoff and distance complexity are depicted as functions of the number of layers. From the figure, we can see that as the number of the layers increases, both complexities increase first and then decrease after reaching a peak. By comparison, we see that C_A's peaks are much more apparent, while C_K is almost flat after reaching the peak. This difference shows that, in the multi-layered structure, the number of spanning trees does not change dramatically with respect to the number of layers, L, if L is not too small (in our case $L > 10$). On the other hand, the distance-related measure, C_A, is sensitive to L.

The convergence rate of a multi-agent group has been shown [73] to be related to the second smallest eigenvalue of the graph Laplacian, $\lambda_2(\mathcal{G})$, which is also called the *algebraic connectivity*. The algebraic connectivity determines how fast the agent group can form a consensus or reach an agreement. The algebraic connectivity of different number of layers are plotted in Figure 37, where sharp peaks can be observed at 8 layers for 120 nodes and 10-12 layers for 240 nodes.

The peaks' location can be interpreted through the information delay of multi-hops needed to propagate information through the graph. When the inter-layer delay and the intra-layer delay are balanced, i.e., when the maximum inter-layer delay is

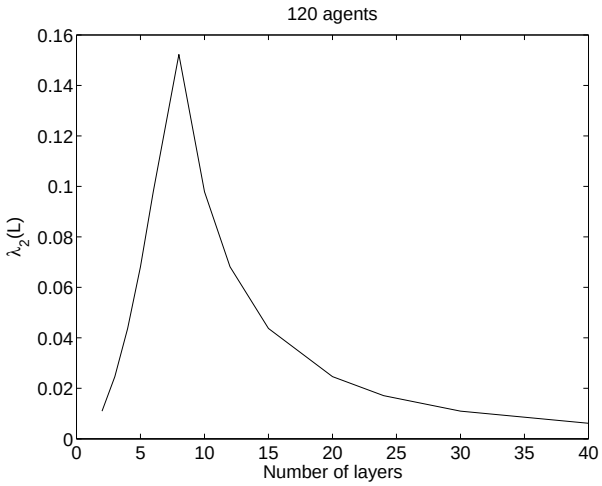


(a) 120 nodes

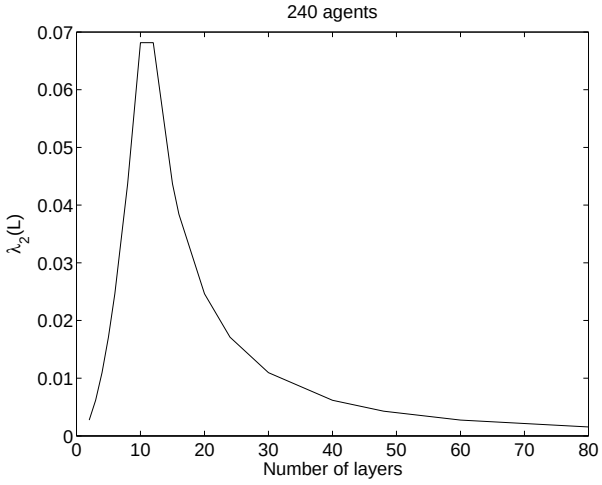


(b) 240 nodes

Figure 36. Complexity measures of the layered structure.



(a) 120 nodes



(b) 240 nodes

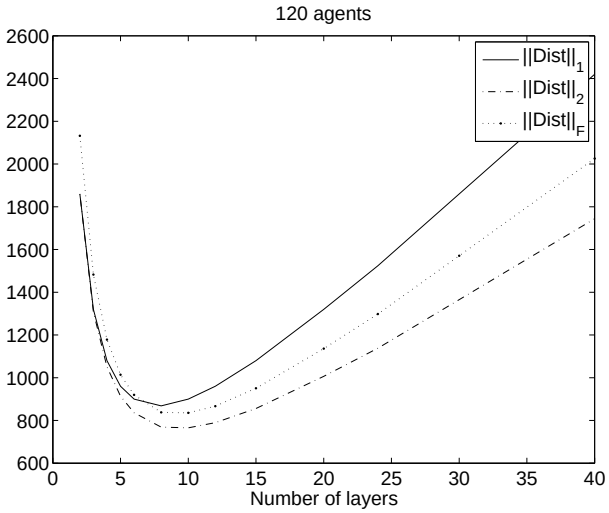
Figure 37. Eigenvalue $\lambda_2(\mathcal{G})$ of the graph Laplacian matrix.

approximately equal to the maximum delay in the same layer, a peak can be observed. In other words, when the longest multi-hop between different layers, $M-1$, is equal or close to the longest multi-hop in each individual layer, $N/2M-1$, a maximal λ_2 can be observed. That is to say that the optimal number of layer is approximately equal to $\sqrt{N/2}$, which is verified by Figure 37.

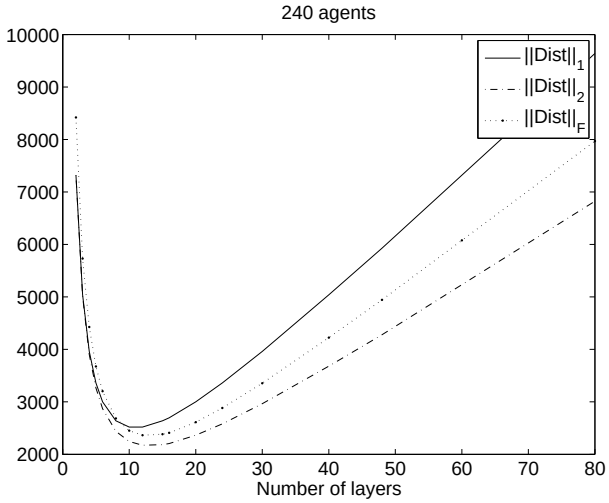
In order to reveal the relationship between the algebraic connectivity and the pairwise distances, we define the distance matrix as $K = [k_{ij}],\ i, j = 1, 2, \ldots, N$. Here i and j are indices running through the node set, and k_{ij} is, as defined before, the length of the shortest path between node i and node j. Different norms of the distance matrices are shown in Figure 38, where we can see that the locations of minimum of the norms, especially the 1-norm, are consistent with the peaks of λ_2. The consistency is not surprising, since λ_2 has been shown to be related to distance invariants of the graph [77]. Further verification of our hypothesis on the algebraic connectivity is left to the future work.

Now let us look at the tradeoff between complexity and the convergence rate. Generally speaking, one would prefer a structure with high convergence rate and low complexity. However, these two properties act in an opposite way in that the fastest convergence rate is associated with the complete graph, which has the highest complexity. On the other hand, the line graph has the lowest complexity [27], while it has the slowest convergence rate. As to our special topology, high complexities also appear in the same range where peaks of convergence rates occur, as shown in Figure 36 and 37. We thus need to choose a structure that balances these two properties.

Different performance indices can be constructed for different situations, where complexity and convergence might have different weights. Here we propose two performance index functions, J_1 and J_2, which combine the consideration of complexity



(a) 120 nodes



(b) 240 nodes

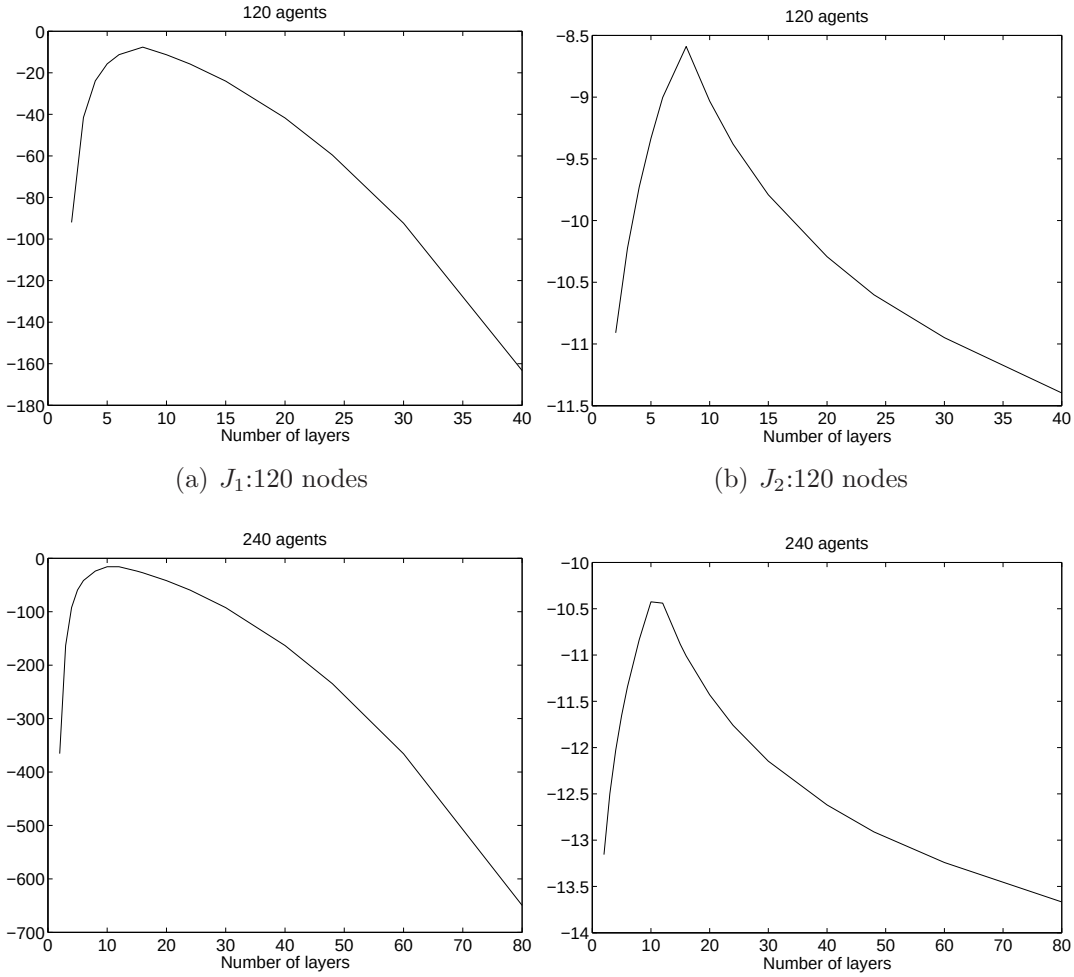
Figure 38. Norms of the distance matrix.

and convergence rate

$$J_1 = -\frac{1}{\lambda_2} - C_K, \tag{111}$$

$$J_2 = \log(\lambda_2/C_A). \tag{112}$$

J_1 utilizes the Kirchhoff complexity, while J_2 uses the distance related complexity. They are depicted in Figure 39, and the optimal structure for 120 nodes (8 layers) is shown in Figure 40.



(a) J_1:120 nodes

(b) J_2:120 nodes

(c) J_1:240 nodes

(d) J_2:240 nodes

Figure 39. Performance indices of the multi-layered multi-agent group.

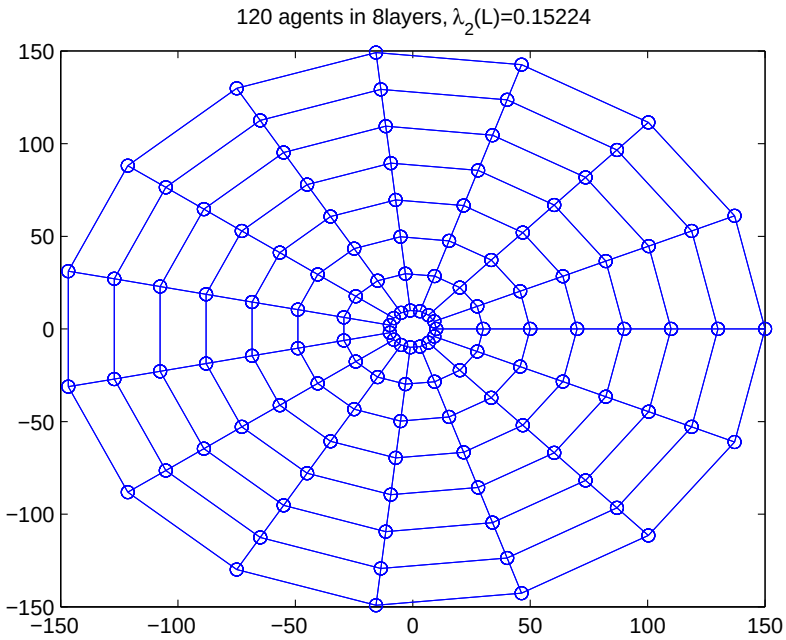


Figure 40. The optimal layered structure for 120 agents: 8-layers.

8.7 Conclusions

In this chapter we present a hybrid Stop-Go control policy for the leaders in a multi-agent containment scenario. In particular, the control strategy allows us to transport a collection of follower-agents to a target area while ensuring that they stay in the convex polytope spanned by the leaders. The enabling results needed in order to achieve this is that, for stationary leaders, the followers in a connected interaction graph will always converge to locations in the leader-polytope. Extensions to the proposed control strategy are given in order to ensure certain liveness properties and we show how the proposed methods lend themselves easily to generalizations to hierarchical information exchange strategies. Examples are presented in order to stress the viability of the proposed approach.

CHAPTER 9

APPLICATIONS

In this chapter, we demonstrate several applications of graph-based control of networked systems. First, we show how the so-called formation switching problem can be solved with the control methods in Chapter 5.2. Then we show how the leader-follower method proposed in Chapter 6 and Chapter 8 can be applied to the problem of optimal control of semi-static equilibrium processes. Finally, we implement the formation control strategy from Chapter 4 on two mobile robots.

9.1 Distributed Formation Switching

In this section, we will show an example of formation switching based on the formation control algorithm proposed in Chapter 4.

Given a multi-agent formation, it can adapt to small perturbations in the environment by the elasticity of the formation, or by a moderate deformation of itself, as shown in Figure 41.

The problem we want to discuss here, however, is how to switch between different formations which is desirable as the environment changes. For example, the flying wedge formation was adopted by both Rome legions and football players for its fierce momentum and penetration ability[1], while the straight line is adopted by migrating birds for energy efficiency. One can also imagine a group of agents engaged in a search-and-rescue operation. A widely spread formation is preferred during the searching phase, while a tightly clustered one, surrounding the rescued object, is preferred as the object is discovered.

In [79], the collaborative beam-forming example was used to show that by adjusting the topology of the network and properly locating each sensor node, maximum

[1]For the same reason, it is banned in the modern football games.

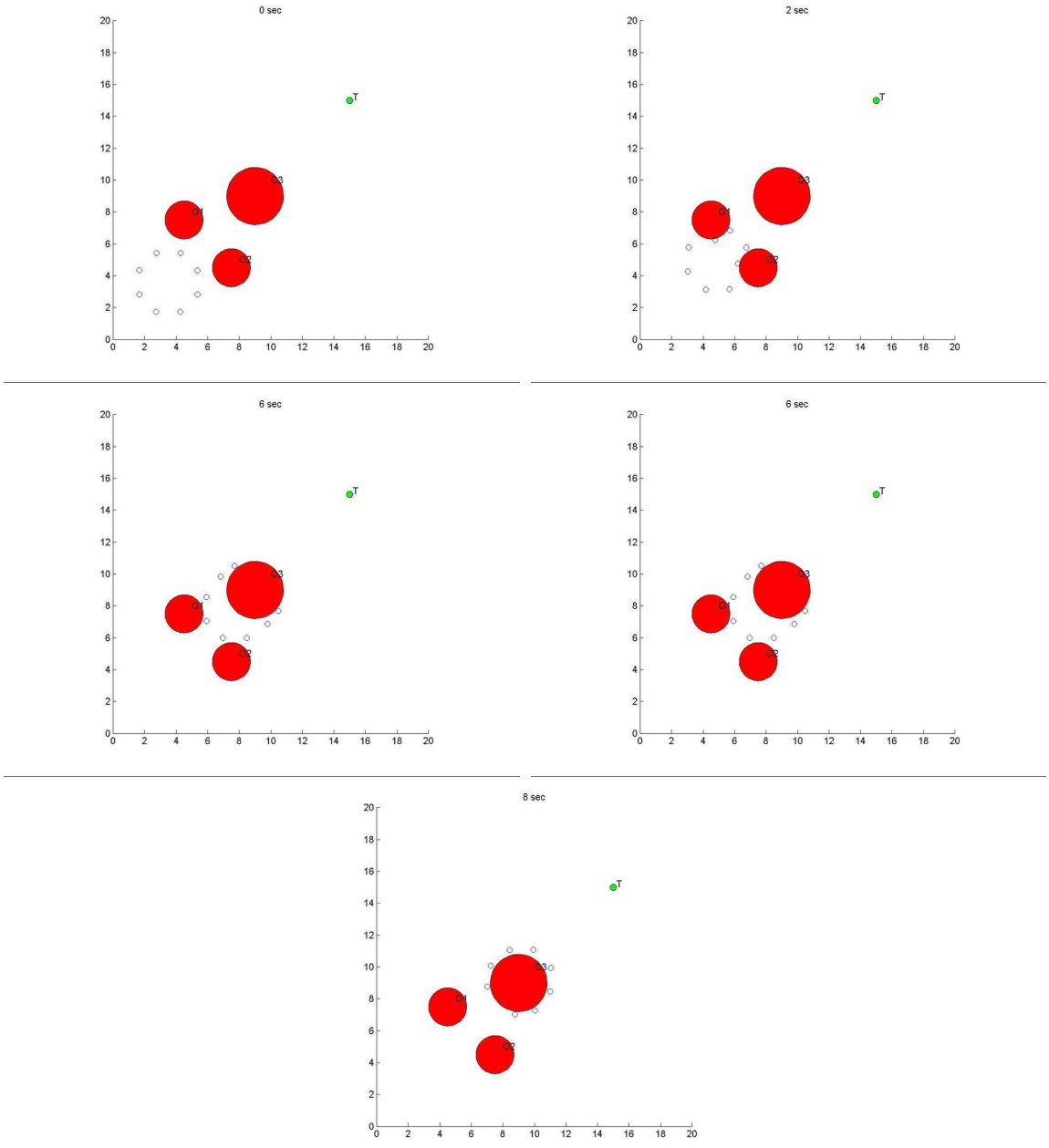


Figure 41. The figure shows an obstacle avoidance behavior, based on formation deformation. The agents are desired to reach the goal in the upper right corner in a circular formation. The group can go through narrow passage or go around not so big obstacles by squeezing or expanding the formation.

signal to interference ratio (SIR) was achieved, as shown in Figure 42. However, the formation-switching strategy proposed in [79] is centralized in the sense that it requires the information of all the agents in order to solve the problem. In addition, only one agent is allowed to move at a certain switching instance.

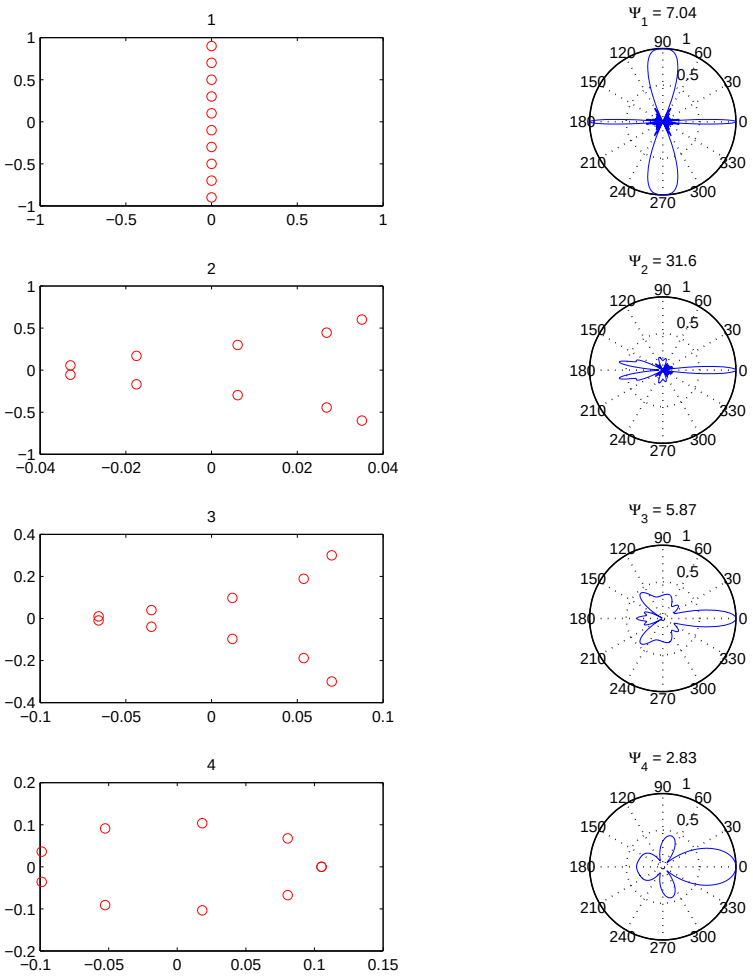


Figure 42. Beam forming performance for various geometries. The power becomes more concentrated in the main lobe (pointing toward 0 degree) as the formation change from a line to a circle.

A reactive formation switching strategy was proposed in [21], where a formation error is generated based on the current environmental conditions and, if necessary, the team switches to the formation with the least error associated with it. The candidates

are a set of formations predefined in the so-called locker room agreement and the switching logic is implemented in a hybrid automaton. Nevertheless, a distributed strategy which is able to disseminate and assemble the information in a localized way is still left at large. In such a strategy, each individual makes its own decisions based on the local information, while all the agents should reach the same final decision.

9.1.1 Leader-Follower Based Formation Switching

In what follows, we also assume that there exists a locker room agreement on what formation to use under given environmental condition. That is to say, given a certain event triggering formation switching, e.g., encountering a narrow corridor, the group has a consensus on which formation should be taken. Moreover, let us assume that at each time only one formation switching event can be triggered and the time between two event is long enough for the group to settle down in a formation.

Given n possible formations, we use $i \in \{1, \ldots, n\}$ to denote one of them. Now, we propose a formation switching strategy based on the leader-follower structure and the controlled agreement problem. If an agent has detected a formation switching event, it will send its decision, the number corresponding to the desired formation, to its neighbor(s) and keep doing so until another even triggered. If an agent has no intention of changing formation but has received the switching information from its neighbor, it will update its state according to the consensus algorithm following the nearest-neighbor rule (3). Recall the control law (48) in the leader-follower structure, it can also be used to describe our proposed decision rule by defining the state of each agent as its choice of formation. The agents getting environmental information as their input are the leaders, while the rest are followers. The leaders are the ones with more knowledge of the environment than the others. Since, we assume there is only one event triggered at each time and the intervals between any two events are long enough, the agents have enough time to settle their decision. The distributed decision making can be viewed as controlled rendezvous with fixed uniform boundary

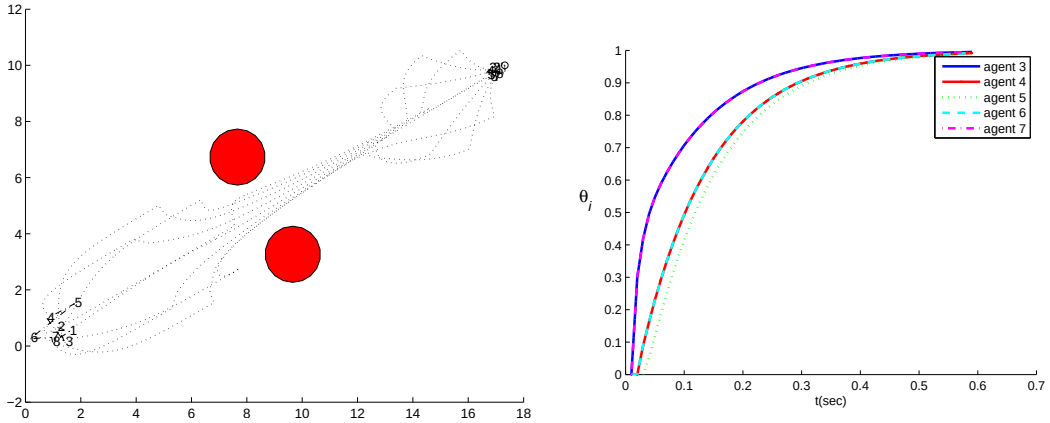
condition,

$$\begin{bmatrix} \dot{\theta}_f \\ \dot{\theta}_l \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_f(\mathcal{G}) & l_{fl}(\mathcal{G}) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \theta_f \\ \theta_l \end{bmatrix}, \tag{113}$$

where $\theta \in [1, n]$ is the state variable corresponding to the choice of formation. From the results in Chapter 6 and Chapter 8, we conclude that a unanimous decision is achievable and all the agents will agree on the choice proposed by the leaders. Notice here, the algorithm guarantee the convergence to a integer, in our example we round θ to the closest integer when the difference between θ and this integer is smaller than a threshold for enough long time.

Next, we use an example to illustrate our method. Imagine a scenario where a group of robots is marching toward a goal behind a narrow corridor. In the free space, a circle formation, denoted by '0', is desired, while a straight line formation, denoted by '1', is needed to pass the corridor. As soon as the front robots encounter the corridor, they stop and send a '1' to their neighbors. On receiving the message, their neighbor start to update their formation index value by the nearest-neighbor rule. At the same time, the agents having received new messages change their motion from formation keeping to rendezvous. Since the leaders (robots with new observations), are static, and the followers (robots without new observation) adopt rendezvous motion, the followers will converge to the convex hull spanned by the leaders as has been proved in Chapter 8. During this procedure, each agent keep observing the difference of the formation index value between itself and its neighbors, if this value is lower than some threshold after some time, it knows an agreement has reached. Now, the leaders also change their mode into rendezvous and the rest part is exactly the same as described in the formation process described in Chapter 5.2. Based on Figure 12, block diagram of our formation switching strategy is shown in Figure 49. Trajectories of the formation switching process is shown in Figure 43, and the snap shots of this process is shown in Figure 44, 45, 46, 47, and 48, where the agents move from the

free space in the lower left corner toward the goal in the upper right corner. They are desire to switch into a straight line in order to go through the narrow corridor along their way.



(a) (b)

Figure 43. The figure shows (a) the trajectory of a formation switching process, and (b) consensus searching process when agent 1,2 and 8 found obstacles.

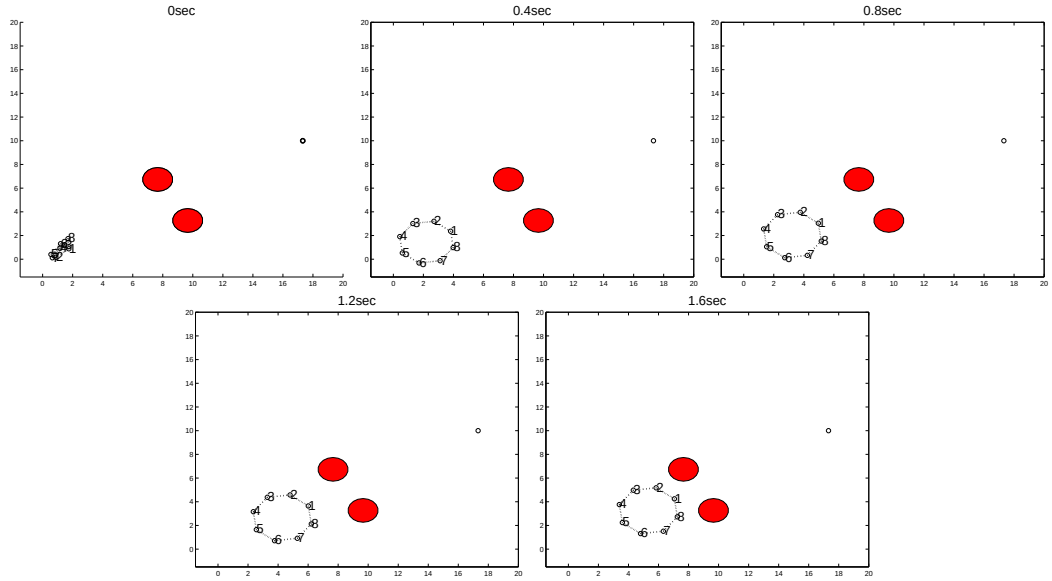


Figure 44. Snap shots of the formation moving in the free space before formation switching is shown in this figure.

9.1.2 Feasible Formation Adjacency Graph and Formation Paths

Based on the distributed decision making strategy (113), we propose a more sophisticated strategy. In a more complicated scenario where many formations might be

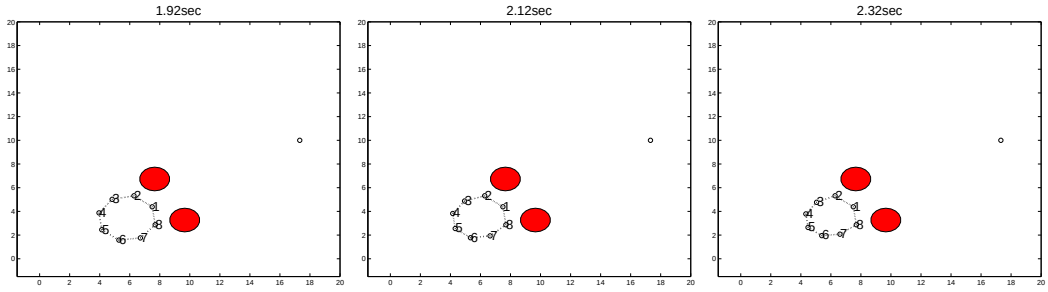


Figure 45. Snap shots of the distributed decision making process are shown in this figure, where agent 1, 2, and 8 take the 'leaders' role and stay put while the rest are moving according to a rendezvous law.

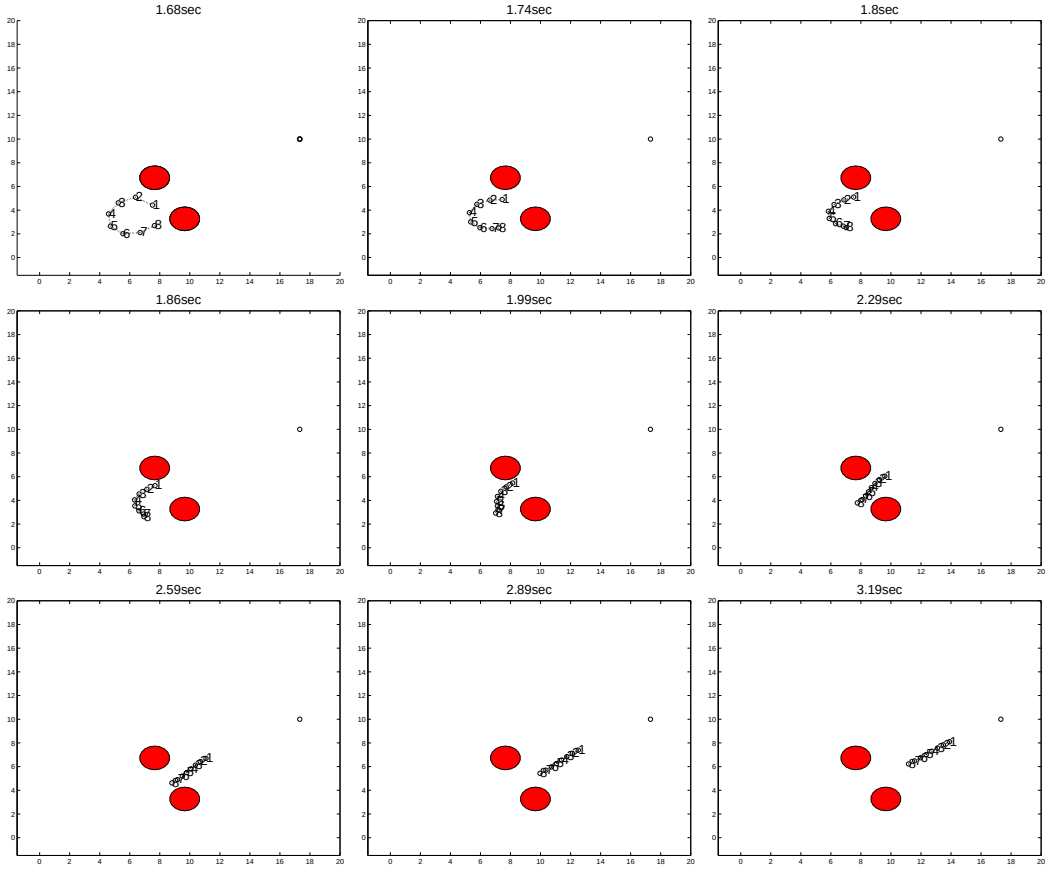


Figure 46. When the agents reaches an consensus on the choice of formation, they start moving from a circular formation to a straight line formation and go through the corridor.

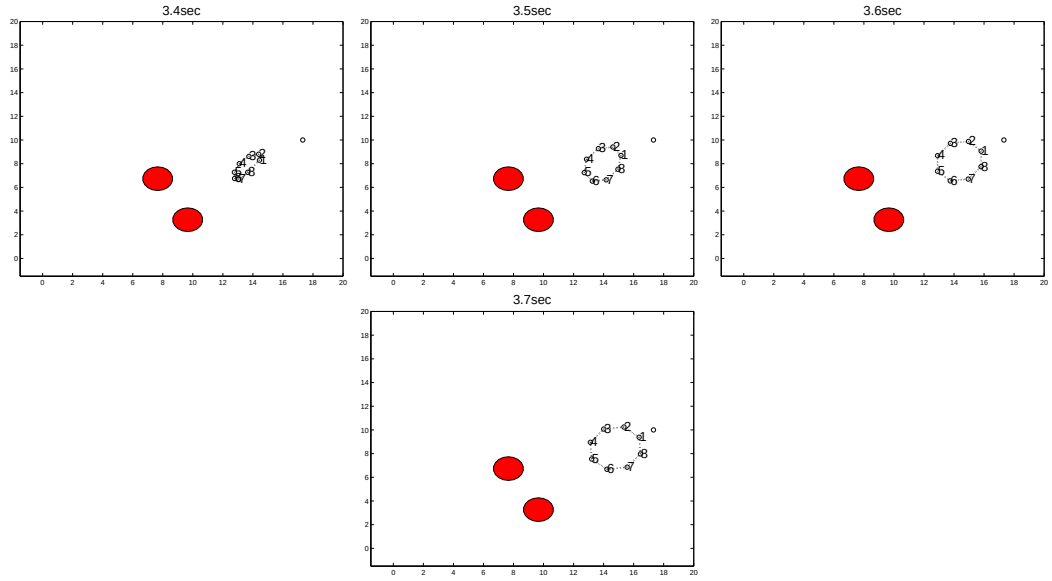


Figure 47. After the agents passed the corridor and move into a free space, they change back into circle. This time agent 8 sends out the message.

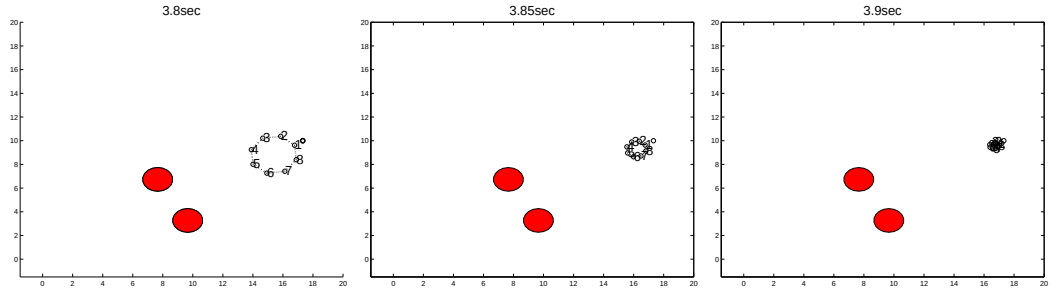


Figure 48. The agents finally rally at the goal point

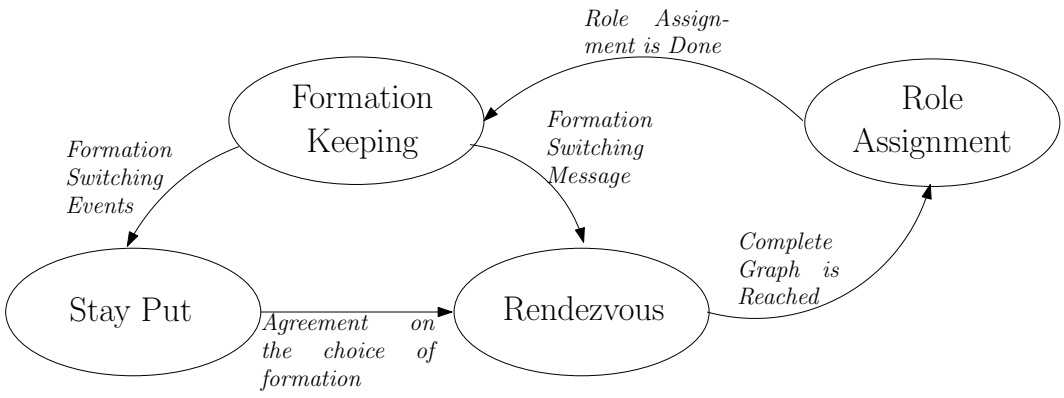


Figure 49. Formation switching process.

involved, the formation process defined in Chapter 5.2 might seem inefficient, since the agents need to gather together to form a complete graph and reassign their roles in the new formation. This procedure might be time and energy consuming. In order to find an more efficient way of formation switching, ideally not gathering involved, we need some more notations.

Two candidate formations are *adjacent* if either of them is a subgraph of the other. The *formation adjacency graph* is a graph, whose node set is the set of all possible formations and whose edge set is determined by the adjacent relations between the possible formations [80].

Since a formation can be encoded as a graph, enhanced with geometrical constraint corresponding to limited sensing range, and we assume the agents have limited sensing range, a formation is not always reachable from an adjacent formation. For example, if formation i and j are adjacent to each other in a formation adjacency graph and $\mathcal{G}_i \subset \mathcal{G}_j$, i is always reachable from formation j by deleting edges but the converse does not hold, since adding edges is subjected to given geometric constraints. By trimming the directions that are not feasible, we get a *feasible formation adjacency graph* which is a induced direction subgraph of formation adjacency graph. The remaining edges, which may be directional, represent the situations where the edges need to be added are geometrically feasible. An example of formation adjacency graph is shown in Figure 50(a).

Now, assume that every formation is reachable from any other formations, i.e., the feasible formation adjacency graph is strongly connected, then switching formation from i to j equals finding a path connecting i to j. We can further augment the feasible formation adjacency graph by associating a cost to each edge as the cost of switching between formations, and thus define a weighted feasible formation adjacency graph. A reasonable cost associated to switching from formation i to j is the number of edges

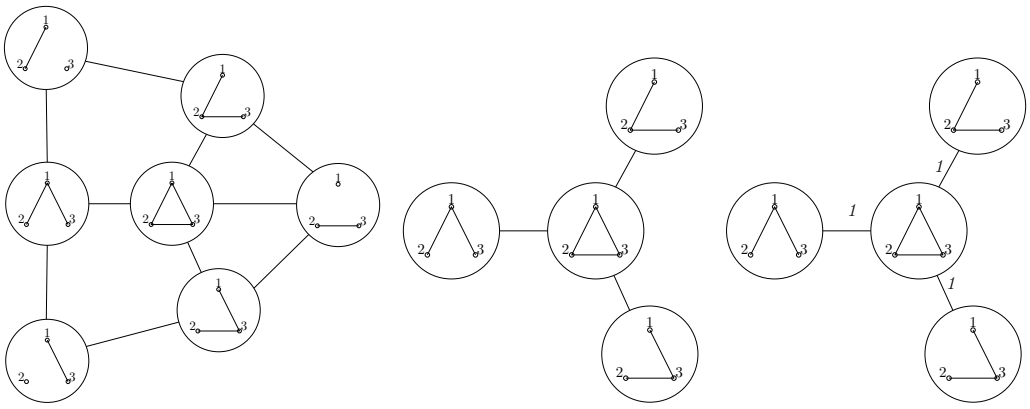


Figure 50. The figure shows (a) a formation adjacency graph of three node formations, and (b) the corresponding feasible formation adjacency graph, and (c) the corresponding weighted feasible formation adjacency graph.

that need to be add or cut from formation i in order to achieve formation j, i.e.,

$$\gamma_{ij} = \left| |E_{\mathcal{G}_i}| - |E_{\mathcal{G}_j}| \right|.$$

A more sophisticated cost might involve the total displacement associated to the switching, but we will not pursuit this topic here.

To this end, the formation switch problem boils down to find a shortest path in the feasible formation adjacency graph that connects the present formation to the desired formation. A weighted feasible formation adjacency graph and an example path are shown in 50(a). Note that the weighted feasible formation adjacency graph and the way of searching for the shortest path is part of the locker room agreement. In the reality, the number of candidate formations is normally small, so the path finding problem will not be very demanding.

At the moment of adding edges, we demand the agents involved to stay put until all the communication and sensing links are established. This can be solved by further trimming the feasible formation adjacency graph in such a way that all the nodes involve in a edges switching are already in a connected subgraph before the switching. So the "ready to go" information need to be sent only in a small connected subgraph. This add a layer of complexity to the feasible formation adjacency graph,

but, this happens in the "locker room" and is carried by the "coach", a machine in charge of off line planning.

Another question needed to be answered is how an agent knows that a formation is achieved. Our answer would be 'keep an eye on your neighbors', if the formation error, the difference between the real displacement and the desired displacement of his neighbors, is lower than some threshold after some time, the agent can assume the formation is achieved. Beyond that, one should add additional dwelling time for the whole formation to settle down.

At this point, we have obtained a strategy, which can solve some formation switching problems. Notice that the purpose of this section is to propose an example of formation switching, but not to achieve a general theoretical framework. Having said that, the idea presented in this section is heuristic to the further study.

9.2 Leader-Based Semi-Static Equilibrium Process

In Chapter 6, we studied the controllability of the leader-follower structure

$$\dot{x}_f = -\mathcal{L}_f(\mathcal{G})x_f - l_{fl}(\mathcal{G})x_l. \tag{114}$$

Results in Chapter 6 enable us to apply optimal control techniques for driving the system between specified positions. It is shown that this problem is in fact equivalent to the problem of driving an invertible linear system between semi-static equilibrium points.

Since we assume the dynamics along each dimension can be decoupled, they can be considered independently, and it is sufficient to analyze the performance along a single dimension. In other words, let $x_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$, be the (one-dimensional) position vector of the ith agent, and let $x = [x_1, x_2, \ldots, x_N]^T$ be the state vector of the group of agents, where N is the total number of agents.

For the sake of notational convenience, we equate x_f with x, and x_l with u throughout this section. Moreover identify A with $-\mathcal{L}_f$ and B with $-l_{fl}$. In Lemma 6.3.2, we

have proved that $\mathcal{L}_f(\mathcal{G})$ is positive definite if $\mathcal{G}$ is connected, therefore A is invertible.

Using this notation, system (114) can be rewritten as

$$\dot{x} = Ax + Bu. \tag{115}$$

Moreover, since the leaders are unconstrained in their motion, let

$$\dot{u} = v,$$

where v is the control input.

For a fixed u, the semi-static equilibrium to (115) is given by

$$x = -A^{-1}Bu. \tag{116}$$

This is just a restatement of Theorem 50.

The problem under consideration here is the semi-static equilibrium process problem, i.e., the problem of transforming (x, u) from an initial point satisfying (116) to a final such point. The equilibrium should be reached in finite time, therefore an optimal control solution will be sought. Define the performance function as follows:

$$J = \frac{1}{2}\int_0^T (\dot{x}^T P\dot{x} + \dot{u}^T Q\dot{u})dt, \tag{117}$$

where $P \succeq 0$ and $Q \succ 0$. The optimal control problem can be formulated as

$$\min_u J. \tag{118}$$

Assuming the system is controllable, we first form the Hamiltonian

$$\begin{aligned} H =& \frac{1}{2}(\dot{x}^T P\dot{x} + \dot{u}^T Q\dot{u}) + \lambda^T(Ax + Bu) + \mu^T v \\ =& \frac{1}{2}(x^T A^T PAx + 2x^T A^T PBu + u^T B^T PBu + v^T Qv) + \lambda^T(Ax + Bu) + \mu^T v, \end{aligned} \tag{119}$$

where λ and μ are the co-states. The first-order necessary optimality condition then gives

$$\begin{aligned}
\frac{\partial H}{\partial v} &= v^T Q + \mu^T = 0 \quad \Rightarrow v = -Q^{-1}\mu, \\
\dot{\lambda} &= -(\frac{\partial H}{\partial x})^T = -A^T PAx - A^T PBu - A^T \lambda, \\
\dot{\mu} &= -(\frac{\partial H}{\partial u})^T = -B^T PAx - B^T PBu - B^T \lambda.
\end{aligned} \tag{120}$$

In other words, by letting $z = [x^T, u^T, \lambda^T, \mu^T]^T$, we obtain the following equation:

$$\dot{z} = Mz, \tag{121}$$

where

$$M = \begin{bmatrix} A & B & 0 & 0 \\ 0 & 0 & 0 & -Q^{-1} \\ -A^T PA & -A^T PB & -A^T & 0 \\ -B^T PA & -B^T PB & -B^T & 0 \end{bmatrix}.$$

Let the initial state be given by

$$z_0 = [x_0^T, u_0^T, \lambda_0^T, \mu_0^T]^T.$$

Now, the problem is to select λ_0 and μ_0 in such a way that, through this choice, we get

$$u(T) = u_T$$

$$x(T) = -A^{-1}Bu_T \triangleq x_T.$$

To achieve this, we partition the matrix exponential in the following way:

$$e^{MT} = \begin{bmatrix} \phi_{xx} & \phi_{xu} & \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{ux} & \phi_{uu} & \phi_{u\lambda} & \phi_{u\mu} \\ \phi_{\lambda x} & \phi_{\lambda u} & \phi_{\lambda\lambda} & \phi_{\lambda\mu} \\ \phi_{\mu x} & \phi_{\mu u} & \phi_{\mu\lambda} & \phi_{\mu\mu} \end{bmatrix}. \tag{122}$$

We can find the initial conditions of the co-states by solving

$$\begin{bmatrix} x_T \\ u_T \end{bmatrix} = \begin{bmatrix} \phi_{xx} & \phi_{xu} & \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{ux} & \phi_{uu} & \phi_{u\lambda} & \phi_{u\mu} \end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \\ \lambda_0 \\ \mu_0 \end{bmatrix}.$$

Now, let

$$\Phi_1 = \begin{bmatrix} \phi_{xx} & \phi_{xu} \\ \phi_{ux} & \phi_{uu} \end{bmatrix} \text{ and } \Phi_2 = \begin{bmatrix} \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{u\lambda} & \phi_{u\mu} \end{bmatrix},$$

then we have

$$\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix} = \Phi_2^{-1} \left(\begin{bmatrix} x_T \\ u_T \end{bmatrix} - \Phi_1 \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \right).$$

Since we are considering a semi-static process, we have

$$x_0 = -A^{-1}Bu_0 \text{ and } x_T = -A^{-1}Bu_T,$$

and hence the initial conditions of the co-states become

$$\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix} = -\Phi_2^{-1}\Psi \begin{bmatrix} u_0 \\ u_T \end{bmatrix},$$

where

$$\Psi = \begin{bmatrix} \phi_{xx}A^{-1}B - \phi_{xu} & -A^{-1}B \\ \phi_{ux}A^{-1}B - \phi_{uu} & I \end{bmatrix}.$$

The invertibility of Φ_2 follows directly from the fact that a particular point-to-point transform always has a unique solution.

As an example, Figure 51 shows a semi-static process, where the dynamics of the system are given by

$$\dot{x} = -x - u,$$

and P and Q are both set to be 1. The system starts from $x_0 = 1, u_0 = -1$ and the desired final position is $x_T = -1, u_T = 1$. The dash-dotted line shows the subspace $\{(x, u) \mid x = -A^{-1}Bu\}$, while the solid line is the actual trajectory of the system under the optimal control law with $T = 2$.

In Figure 52, the snapshots of a herding process are shown where the leaders (black) maneuver the followers (white) from an initial position to a desired final position in finite time.

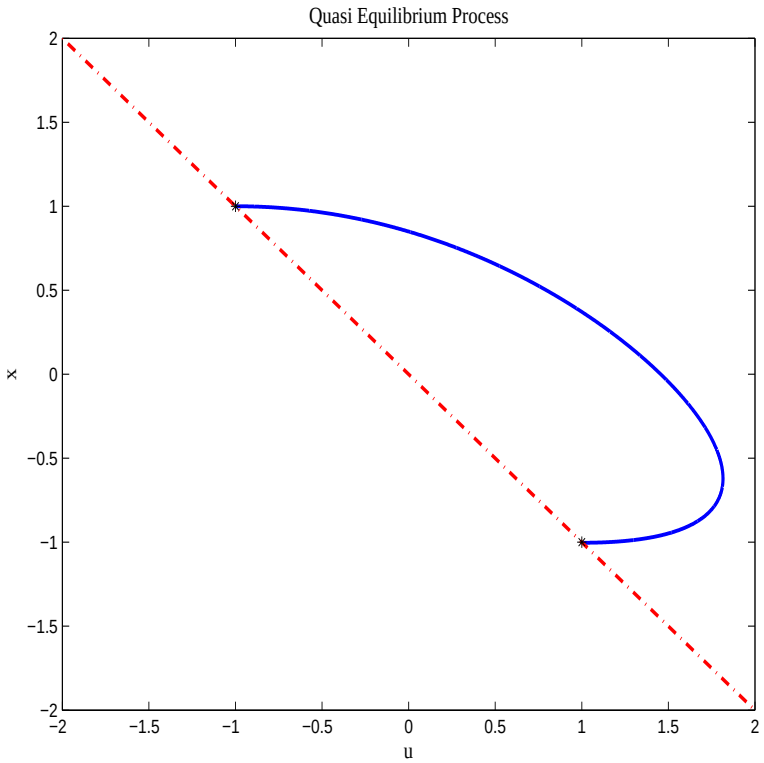


Figure 51. An example of semi-static process for the system $\dot{x} = -x - u$, $P = Q = 1$, $x_0 = 1, u_0 = -1, x_T = -1, u_T = 1$.

9.3 Robotics Implementation

Multi-agent robotics is a promising platform on which many networked system theories can be applied. In order to verify the algorithm proposed in Chapter 4, the formation control strategy is implemented on two Magellan Pro mobile robots in the GRITS lab at the Georgia Institute of Technology.

The Magellan Pro is a commercially available indoor mobile robot. The onboard computer is currently running Fedora Linux, and each robot has 16 sonars, 16 IR sensors, and a pan-tilt-zoom (PTZ) camera. It provides us with a computationally capable and sensor-rich platform for mobile robotics. Player, a flexible device server, is installed on top of the hardware to provide interface to a variety of sensors and actuators.

In order to realize our algorithm, we need to develop our client program using the functionality provided by the player server to interact with the robot. In our

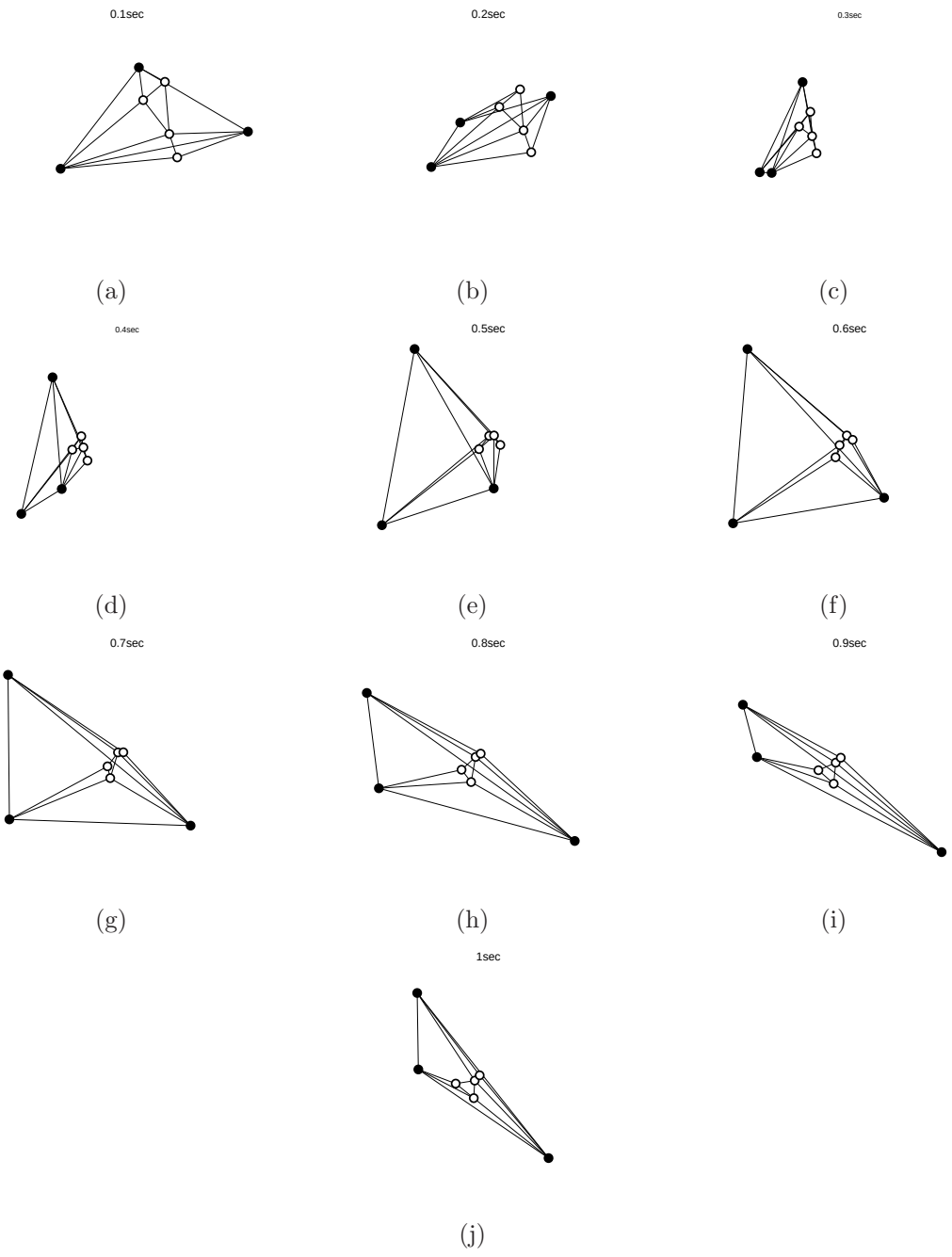


Figure 52. A semi-static process where 3 leaders (black nodes) herd 4 followers (white nodes), where $T = 1$ *sec*.

experiments, we used sonar sensors for distance keeping and collision avoidance, and the PTZ camera for keeping the bearing angle.

Given the pair of robots available, we design two experiments. The first task, shown in Figure 53(a) is to implement a line formation, with person was holding a green blob and acting as the leader. *Robot 1* follows the green blob with the on board PTZ camera, and *Robot 2* follows the yellow blob on the first robot. In the second experiment, a triangle formation is desired, where the leader on the top vertex and *Robot 1* follows the leader from behind and Robot 2 follows the leader with an angle, as shown in Figure 53(b).

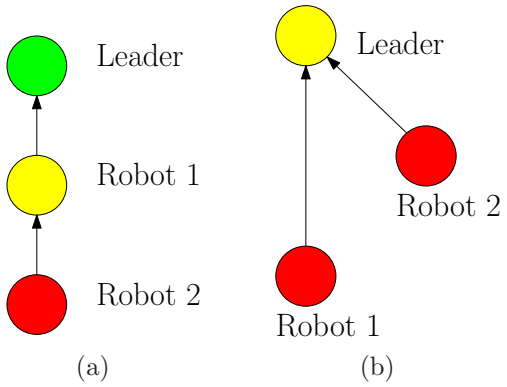


Figure 53. The figure shows desired formations (a) a line formation where the *Robot 1* follows the green blob (the team leader), and *Robot 2* follows *Robot 1*, and (b) a triangle formation where both robots follow the yellow blob (the leader).

In the real implementation, we adopt a behavior based method. The input to the robot, i.e., desired linear and angular velocity, is a weighted combination several components. Each component represents a behavior, e.g. go-to-goal, collision avoidance, formation. The weight of each behavior is determined by the importance or urgency of the behavior. The experiment results are shown in Figure 54 and Figure 55. The formation control strategy from Chapter 4 is proved to be effective in coordinating robot groups.

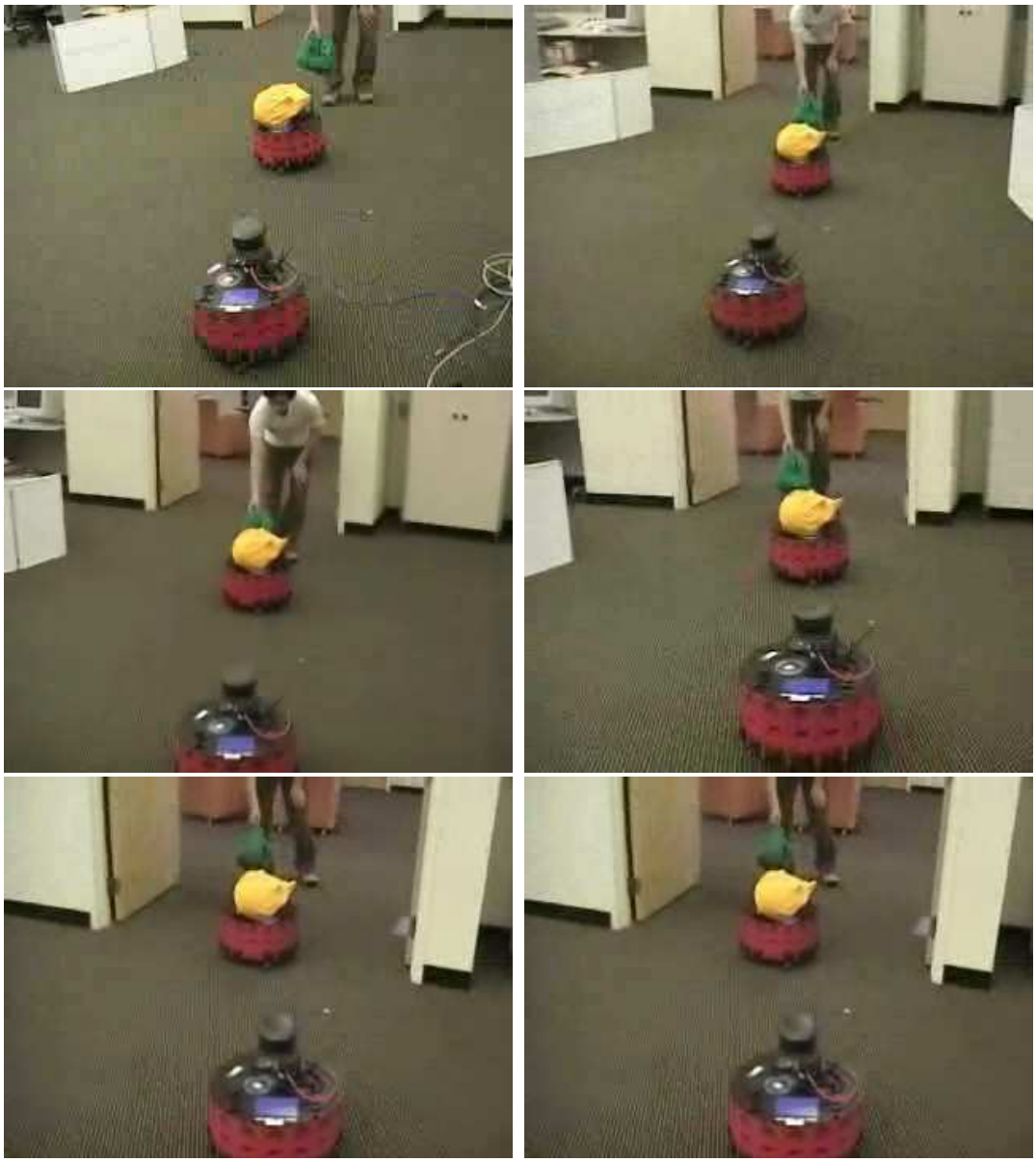


Figure 54. The figure shows a line formation where the *Robot 1* is following the green blob (the team leader).

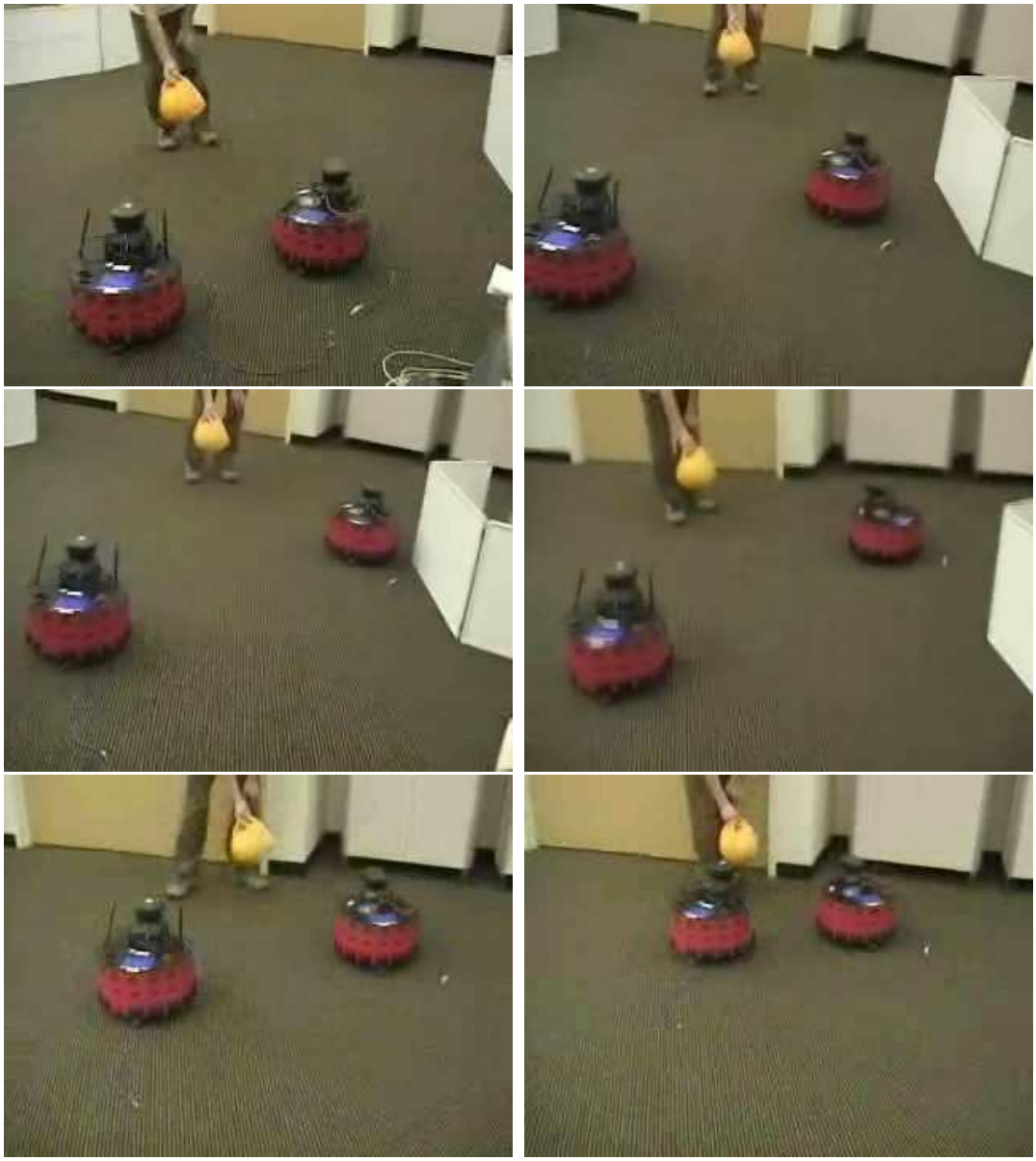


Figure 55. The figure shows a triangle formation where both robots are following the yellow.

REFERENCES

[1] C. Reynolds, "Flocks, herds and schools: a distributed behavioral model," in *Computer Graphics (ACM SIGGRAPH '87 Conference Proceedings)*, vol. 21(4), pp. 25–34, July 1987.

[2] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226–1229, 1995.

[3] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 818–828, Oct. 1999.

[4] J. A. Fax and R. M. Murray, "Graph laplacian and stabilization of vehicle formations," in *Proceedings of the 15th IFAC Conf*, pp. 283–288, 2002.

[5] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465–1476, Sept 2004.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, June 2003.

[7] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[8] W. Ren and R. Beard, "Consensus of information under dynamically changing interaction topologies," in *Proceedings of the American Control Conference 2004*, vol. 6, pp. 4939–4944, June 30-July 2 2004.

[9] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, March 2006.

[10] R. Olfati-Saber and R. M. Murray, "Distributed structural stabilization and tracking for formations of dynamic multi-agents," in *Proceedings of the 41st IEEE Conference on Decision and Control 2002*, vol. 1, pp. 209–215, Dec. 2002.

[11] R. Olfati-Saber and R. M. Murray, "Flocking with obstacl avoidance: cooperation with limited communication in mobile networks," in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control 2003*, vol. 2, (Maui, Hawaii USA), pp. 2022–2028, Dec. 2003.

[12] R. Olfati-Saber and R. M. Murray, "Agreement problems in networks with directed graphs and switching toplogy," in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control 2003*, vol. 4, (Maui, Hawaii USA), pp. 4126–4132, Dec. 2003.

[13] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part II : Dynamic topology," in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, pp. 2016–2021, 2003.

[14] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in d dimension," *IEEE Transactions on Robotics and Automation*, vol. 51, no. 8, pp. 1289–1298, 2006.

[15] M. Ji and M. Egerstedt, "Connectedness preserving distributed coordination control over dynamic graphs," in *Proceedings of the American Control Conference 2005*, (Portland, OR), pp. 93–98, June 2005.

[16] D. Spanos and R. Murray, "Robust connectivity of networked vehicles," in *Proceeding of the 43rd IEEE Conference on Decision and Control 2004*, vol. 3, pp. 2893–2898, Dec. 2004.

[17] M. Zavlanos and G. Pappas, "Controlling connectivity of dynamic graphs," in *Proceeding of the 44th IEEE Conference on Decision and Control 2005 and 2005 European Control Conference (CDC-ECC '05)*, pp. 6388–6393, Dec. 2005.

[18] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, Dec. 1998.

[19] J. Lawton, R. Beard, and B. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 933–941, Dec. 2003.

[20] M. Egerstedt and X. Hu, "A hybrid control approach to action coordination for mobile robots," *Automatica*, vol. 38, pp. 125–130, Jan. 2002.

[21] H. Axelsson, M. Abubakr, and M. Egerstedt, "Autonomous formation switching for multiple, mobile robots," in *IFAC Conference on Analysis and Design of Hybrid Systems*, (Saint-Malo, Brittany, France), june 2003.

[22] K. Sugihara and I. Suzuki, "Distributed motion coordination of multiple robots," in *Proceedings of IEEE Int. Symp. on Intelligent Control*, pp. 138–143, 1990.

[23] M. Mesbahi, "On a dynamic extension of the theory of graphs," in *Proceedings of the 2002 American Control Conference*, vol. 2, pp. 1234–1239, May 2002.

[24] M. Mesbahi, "State-dependent graphs," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, (Maui, Hawaii USA), pp. 3058–3063, Dec. 2003.

[25] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, vol. 48, pp. 692–697, April 2003.

[26] Y. Liu, K. M. Passino, and M. M. Polycarpou, "Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 76–95, 2003.

[27] A. Muhammad and M. Egerstedt, "On the structural complexity of multi-agent robot formations," in *American Control Conference 2004*, (Boston, MA), 2004.

[28] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, pp. 116–120, Jan. 2006.

[29] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc.IEEE Int. Conf. Robot. Automat.*, (Leuven, Belgium), pp. 2864–2869, May 1998.

[30] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 905–908, Dec. 2001.

[31] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 947–951, Dec. 2001.

[32] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control*, vol. 46, pp. 1777–1782, Nov. 2001.

[33] P. Ögren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 847–851, Oct. 2002.

[34] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the IEEE Conference on Decision and Control 2001*, (Orlando, Florida), pp. 2968–2973, Dec. 2001.

[35] R. W. Beard, J. R. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, pp. 777–790, Nov. 2001.

[36] T. Eren, P. Belhumeur, B. Anderson, and A. Morse, "A framework for maintaining formations based on rigidity," in *Proceedings of the 2002 IFAC Congress*, pp. 2752–2757, 2002.

[37] H. Tanner, G. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 443–455, June 2004.

[38] L. E. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 681–689, 2003.

[39] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Journal of Robotics and Autonomous Systems*, vol. 31, no. 6, pp. 109–117, 2000.

[40] G. Ferrari-Trecate, M. Egerstedt, A. Buffa, and M. Ji, *Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control.* Santa Barbara, CA: Springer-Verlag, March 2006.

[41] M. Ji, M. Egerstedt, G. Ferrari-Trecate, and A. Buffa, "Hierarchical containment control in heterogeneous mobile networks," in *in Proceedings of Mathematical Theory of Networks and Systems*, (Kyoto, Japan), July 2006.

[42] H. G. Tanner, "On the controllability of nearest neighbor interconnections," in *Proceedings of the* 43^{rd} *IEEE Conference on Decision and Control*, pp. 2467–2472, Dec 2004.

[43] A. Rahmani and M. Mesbahi, "On the controlled agreement problem," in *Proceedings of the American Control Conference 2006*, pp. 1376–1381, June 2006.

[44] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," in *Proceedings of the American Control Conference 2006*, (Minneapolis, MN), pp. 1358–1363, June 2006.

[45] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems: from a graph-theoretic perspective." Submitted to SIAM Journal on Control and Optimization, Special Issue on Networked Controlled Systems, 2007.

[46] V. Gupta, *Distributed Estimation and Control in Networked Systems*. PhD thesis, California Institute of Technology, June 2006.

[47] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *IEEE Computer*, vol. 37, pp. 41–49, Aug 2004.

[48] S. M. Brennan, A. M. Mielke, and D. C. Torney, "Radioactive source detection by sensor networks," *IEEE Transactionas on Nuclear Science*, vol. 52, June 2005.

[49] P. Ting and R. Iltis, "Multitarget motion analysis in a dsn," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, pp. 1125–1139, Sept.-Oct. 1991.

[50] J. Chen, K. Yao, and R. Hudson, "Source localization and beamforming," *Signal Processing Magazine, IEEE*, vol. 19, pp. 30–39, Mar. 2002.

[51] Z. Jin and R. M. Murray, "Multi-hop relay protocols for fast consensus seeking," in *45th IEEE Conference on Decision and Control*, (San Diego, California), 13-15, Dec, 2006.

[52] H. Qi, S. Iyengar, and K. Chakrabarty, "Multiresolution data integration using mobile agents in distributed sensor networks," *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, vol. 31, pp. 383–391, Aug 2001.

[53] J. Kurhinen, "Mp2p network in collecting data from sensor networks," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, pp. 26–29, June 2006.

[54] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference*, Dec. 2005.

[55] V. Gupta, D. Jeffcoat, and R. M. Murray, "On sensor coverage by mobile sensors," in *Proceedings of the IEEE Conference on Decision and Control*, Dec 2006.

[56] V. Gupta, A. F. Dana, J. P. Hespanha, and R. M. Murray, "Data transmission over networks for estimation," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, (MTNS 06)*, June 2006.

[57] C. Godsil and G. Royle, *Algebraic graph theory*. Springer, 2001.

[58] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem," in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, (Maui, Hawaii USA), pp. 1508–1513, Dec. 2003.

[59] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169–182, Feb. 2005.

[60] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *Proceedings of the 2004 American Control Conference*, (Minneapolis, MN), pp. 2124–2129, June 2006.

[61] J.-M. McNew and E. Klavins, "Locally interacting hybrid systems with embedded graph grammars," in *IEEE Conference on Decision and Control*, (San Diego, CA), Dec. 2006.

[62] A. Muhammad and M. Egerstedt, "Connectivity graphs as models of local interactions.," *Journal of Applied Mathematics and Computation*, vol. 168, pp. 243–269, Sept. 2005.

[63] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly 2*, pp. 83–97, 1955.

[64] K. Ogata, *Modern Control Engineering*. Prentice Hall, 4 ed., 2001.

[65] D. Luenberger, *Optimization by vector space methods*. Wiley, 1969.

[66] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in fixed and switching networks." To appear on IEEE Transactions on Automatic Control.

[67] R. Olfati-Saber, "A unified analytical look at Reynolds flocking rules," Tech. Rep. CIT-CDS 03-014, California Institute of Technology, 2003.

[68] M. Egerstedt and C. Martin, "Conflict resolution for autonomous vehicles: A case study in hierarchical control design.," *International Journal of Hybrid Systems*, vol. 2, no. 3, pp. 221–234, 2002.

[69] H. Sussmann, "A maximum principle for hybrid optimal control problems," in *38th IEEE Conference on Decision and Control*, 1999.

[70] A. Bensoussan and J.-L. Menaldi, "Difference equations on weighted graphs," *Journal of Convex Analysis (Special issue in honor of Claude Lemaréchal)*, vol. 12, no. 1, pp. 13–44, 2005.

[71] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multiple agents formations through Partial difference Equations," Tech. Rep. 5-PV, IMATI-CNR, 2004.

[72] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multiagent systems through partial difference equations. Part I: The Laplacian control," in *16th IFAC World Congress on Automatic Control*, 2005.

[73] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans on Autom. Control*, vol. 49, no. 9, pp. 101–115, 2004.

[74] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multiagent systems through partial difference equations," *IEEE Trans. on Automatic Control*, vol. 51, no. 6, pp. 1058–1063, 2006.

[75] R. Dautray and J.-L. Lions, *Mathematical analysis and numerical methods for science and technology. Vol. 5-6: Evolution problems I-II*. Berlin: Springer-Verlag, 1992.

[76] K. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry., "Regularization of zeno hybrid automata," *Systems and Control Letters*, vol. 38, no. 6, pp. 141–150, 1999.

[77] B. Mohar, "The laplacian spectrum of graphs," in *Sixth International Conference on the Theory and Applications of Graphs*, pp. 871–898, 1988.

[78] B. Bollobás, *Modern graph theory*. Graduate texts in Mathematics, Springer-Verlag, 1998.

[79] A. Muhammad, M. Ji, and M. Egerstedt, "Applications of connectivity graph processes in networked sensing and control," in *Networked Embdedded Sensing and Control, Lecture Notes in Control and Information Sciences (LNCIS),* ,, vol. 331, Springer, 2006.

[80] J. Hu and S. Sastry, "Optimal collision avoidance and formation switching on riemannian manifolds," in *in Proc. 40th IEEE Int. Conf. Decision and Control*, vol. 2, (Orlando, FL,), pp. 1071–1076,, 2001.